µPAC-5000 User Manual
(C Language Based)

Version 1.0.1, June 2010

Service and usage information for

µPAC-5001 (D) / µPAC-5001P (D)

µPAC-5001-FD (D) / µPAC-5001-FD (D)

µPAC-5002 (D) / µPAC-5002P (D)

µPAC-5002-FD (D) / µPAC-5002P-FD (D)

µPAC-5002-NV (D) / µPAC-5002P-NV (D)

µPAC-5002-SM (D) / µPAC-5002P-SM (D)

Warranty

All products manufactured by ICP DAS are under warranty regarding defective materials for a period of one year, beginning from the date of delivery to the original purchaser.

Warning

ICP DAS assumes no liability for any damage resulting from the use of this product.ICP DAS reserves the right to change this manual at any time without notice. The information furnished by ICP DAS is believed to be accurate and reliable. However, no responsibility is assumed by ICP DAS for its use, not for any infringements of patents or other rights of third parties resulting from its use.

Copyright

Copyright @ 2009 by ICP DAS Co., Ltd. All rights are reserved.

Trademark

The names used for identification only may be registered trademarks of their respective companies.

Contact US

If you have any problem, please feel free to contact us.
You can count on us for quick response.

Email: service@icpdas.com

Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

Table of Contents

1. Introduction ---------- 6
  1.1. µPAC-5000 (C Language Based) ---------- 7
    1.1.1. µPAC-5000 Release Module ---------- 7
      1.1.1.1. General Series ---------- 7
      1.1.1.2. GPS Series ---------- 7
      1.1.1.3. GPRS Series ---------- 7
      1.1.1.4. GPS & GPRS Series ---------- 8
    1.1.2. µPAC-5000 Module Naming Convention ---------- 9
    1.1.3. µPAC-5000 Comparison ---------- 10
      1.1.3.1. General Series ---------- 10
      1.1.3.2. GPS Series ---------- 11
      1.1.3.3. GPRS Series ---------- 12
      1.1.3.4. GPS & GPRS Series ---------- 13
  1.2. Features ---------- 14
  1.3. Specifications ---------- 18
    1.3.1. General Series ---------- 18
    1.3.2. GPS Series ---------- 19
    1.3.3. GPRS Series ---------- 19
    1.3.4. GPS & GPRS Series ---------- 19
  1.4. Overview ---------- 20
  1.5. Dimensions ---------- 24
  1.6. Companion CD ---------- 25

2. Getting Started ---------- 26
  2.1. Hardware Installation ---------- 27
  2.2. Software Installation ---------- 30

Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

2.3. Boot Configuration ---------- 32
2.4. Uploading µPAC-5000 Programs ---------- 33
2.4.1. Establishing a connection between PC and µPAC-5000 ---------- 34
2.4.1.1. RS-232 connection ---------- 35
2.4.1.2. Ethernet Connection ---------- 38
2.4.2. Uploading and executing µPAC-5000 programs ---------- 43
2.4.3. Making programs start automatically ---------- 44
2.5. Updating µPAC-5000 OS image ---------- 46

3. "Hello World" - Your First Program ---------- 49
3.1. C Compiler Installation ---------- 50
3.1.1. Installing the C compiler ---------- 51
3.1.2. Setting up the environment variables ---------- 55
3.2. µPAC-5000 APIs ---------- 58
3.3. First Program in µPAC-5000 ---------- 59

4. APIs and Demo References ---------- 70
4.1. API for COM Port ---------- 74
4.1.1. Types of COM port functions ---------- 75
4.1.2. API for MiniOS7 COM port ---------- 76
4.1.3. API for standard COM port ---------- 79
4.1.4. Port functions Comparison ---------- 82
4.1.5. Request/Response protocol define on COM port ---------- 84
4.2. API for I/O Modules ---------- 85
4.3. API for EEPROM ---------- 87
4.4. API for Flash Memory ---------- 89
4.5. API for NVRAM ---------- 91
4.6. API for 5-Digital LED ---------- 93

Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

4.7. API for Timer ---------- 95
4.8. API for WatchDog Timer (WDT) ---------- 97
4.9. API for MFS (For µPAC-5000-FD series only) ---------- 99
4.10. API for microSD ---------- 105

Appendix A. What is MiniOS7? ---------- 110

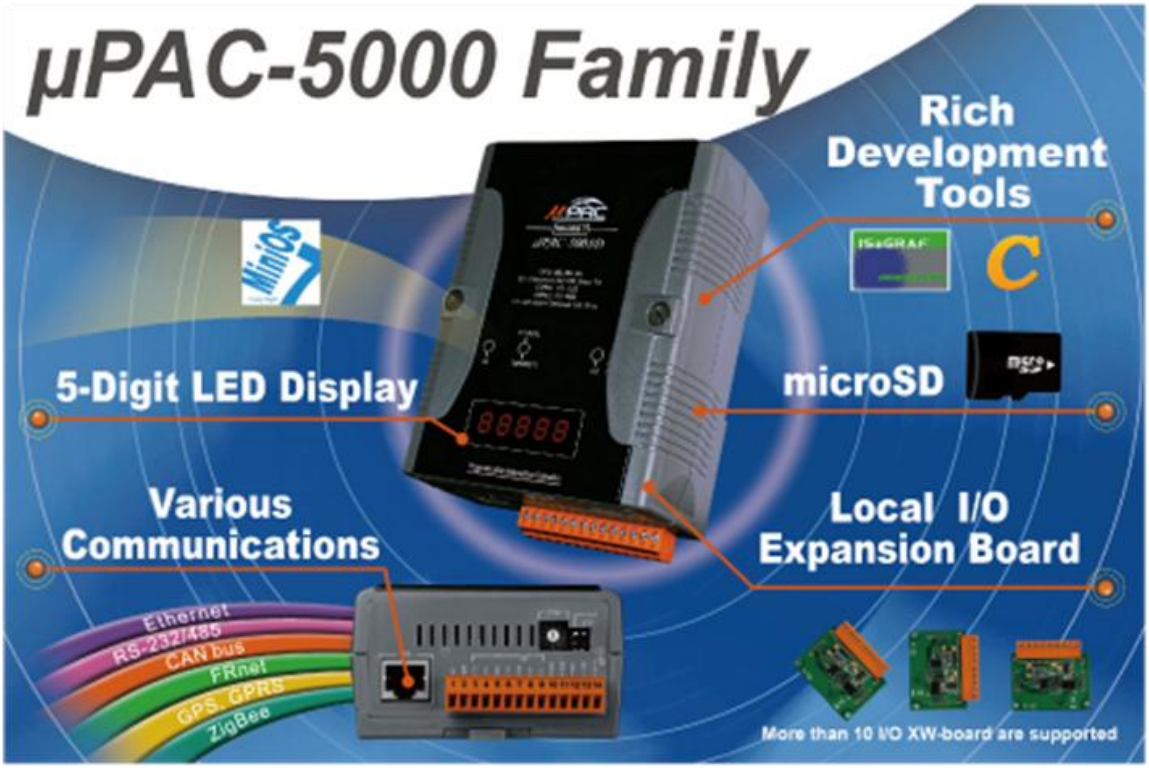
Appendix B. What is MiniOS7 Utility? ---------- 111

Appendix C. What is MiniOS7 File System (MFS)? ---------- 112

Appendix D. More C Compiler Settings ---------- 115
D.1. Turbo C 2.01 ---------- 116
D.2. BC++ 3.1. IDE ---------- 119
D.3. MSC 6.00 ---------- 123
D.4. MSVC 1.50 ---------- 125

Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

1. Introduction



µPAC-5000 family is a palm size PAC. It can install in narrow space and survive in harsh environment. For different applications, µPAC-5000 family offers 3 series, µPAC-5000, LP-5000 and WP-5000. Customers who needs MiniOS7, Linux, or WinCE , all can find a product to fit their requirement..

µPAC-5000 family offers two CPU types (80186, PAX270), 3 OS solutions (WinCE 5.0, Linux, MiniOS7) and several software development toolkits (C, VS .NET, ISaGRAF, InduSoft) for chosen, and all of them are featured same stability and flexibility as ICP DAS's PAC family. This makes µPAC-5000 have a good potential to apply to factory automation, building automation, machine automation, manufacturing management, environment monitoring, etc.

Besides, combining one of the optional expansion boards (XW-boards) for µPACs, such as DI, DO, A/D, D/A, Timer/Counter, communication interface (RS-232/422/485, CAN, FRnet), etc., customers can customize the hardware specification of their µPAC-5000 for different situations.

Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

1.1. µPAC-5000 (C Language Based)

1.1.1. µPAC-5000 Release Module

The µPAC-5000 Series are divided into the General series, GPS series and GPRS series.

1.1.1.1. General Series

µPAC-5000 can be divided into five types, according to their features.

- µPAC-5001/µPAC-5001P
- µPAC-5001-FD/µPAC-5001P-FD
- µPAC-5002/µPAC-5002P
- µPAC-5002-FD/µPAC-5002P-FD
- µPAC-5002-NV/µPAC-5002P-NV
- µPAC-5002-SM/µPAC-5002P-SM

For their comparison, please refer to section “1.1.3.1. General Series”

1.1.1.2. GPS Series

GPS Series coming soon.
Please visit our products page or contact us for more information.
http://www.icpdas.com/products/PAC/up-5000/selection_guide.htm

1.1.1.3. GPRS Series

GPRS Series coming soon.
Please visit our products page or contact us for more information.
http://www.icpdas.com/products/PAC/up-5000/selection_guide.htm

Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

1.1.1.4. GPS & GPRS Series

GPS & GPRS Series coming soon.

Please visit our products page or contact us for more information.

http://www.icpdas.com/products/PAC/up-5000/selection_guide.htm

Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

1.1.2. µPAC-5000 Module Naming Convention

As you examine this manual, you'll notice there are many different products available. Sometimes it is difficult to remember the specifications for any given product. However, if you take a few minutes to understand the module naming conventions, it may save you some time and confusion. The figure below shows how the module naming conventions work for eachµPAC-5000 product.

Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

1.1.3. µPAC-5000 Comparison

1.1.3.1. General Series

The following table provides a comparison of µPAC-5000 General Series specifications

Model	CPU	Flash	SRAM	Memory Expansion	PoE	GPS	GPRS
µPAC-5001 µPAC-5001P	80 MHz	512 KB	512 KB	microSD	- Yes	-	-
µPAC-5001-FD µPAC-5001P-FD				microSD + 64 MB Flash	- Yes		
µPAC-5002 µPAC-5002P			768 KB	microSD	- Yes		
µPAC-5002-FD µPAC-5002P-FD				microSD + 64 MB Flash	- Yes		
µPAC-5002-NV µPAC-5002P-NV				microSD + 1 MB NVRAM	- Yes		
µPAC-5002-SM µPAC-5002P-SM				microSD + 512 KB Battery Backup SRAM	- Yes		

Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

1.1.3.2. GPS Series

The following table provides a comparison of µPAC-5000 GPS Series specifications

Model	CPU	Flash	SRAM	Memory Expansion	PoE	GPS	GPRS
µPAC-5101 µPAC-5101P	80 MHz	512 KB	512 KB	microSD	- Yes	Yes	-
µPAC-5101-FD µPAC-5101P-FD				microSD + 64 MB Flash	- Yes		
µPAC-5102 µPAC-5102P			768 KB	microSD	- Yes		
µPAC-5102-FD µPAC-5102P-FD				microSD + 64 MB Flash	- Yes		
µPAC-5102-NV µPAC-5102P-NV				microSD + 1 MB NVRAM	- Yes		
µPAC-5102-SM µPAC-5102P-SM				microSD + 512 KB Battery Backup SRAM	- Yes		

Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

1.1.3.3. GPRS Series

The following table provides a comparison of µPAC-5000 GPRS Series specifications

Model	CPU	Flash	SRAM	Memory Expansion	PoE	GPS	GPRS
µPAC-5201	80 MHz	512 KB	512 KB	microSD	-	-	Yes
µPAC-5201-FD				microSD + 64 MB Flash	-		
µPAC-5202			768 KB	microSD	-		
µPAC-5202-FD				microSD + 64 MB Flash	-		
µPAC-5202-NV				microSD + 1 MB NVRAM	-		
µPAC-5202-SM				microSD + 512 KB Battery Backup SRAM	-		

Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

1.1.3.4. GPS & GPRS Series

The following table provides a comparison of µPAC-5000 GPS & GPRS Series specifications

Model	CPU	Flash	SRAM	Memory Expansion	PoE	GPS	GPRS
µPAC-5301	80 MHz	512 KB	512 KB	microSD	-	Yes	Yes
µPAC-5301-FD				microSD + 64 MB Flash	-		
µPAC-5302			768 KB	microSD	-		
µPAC-5302-FD				microSD + 64 MB Flash	-		
µPAC-5302-NV				microSD + 1 MB NVRAM	-		
µPAC-5302-SM				microSD + 512 KB Battery Backup SRAM	-		

Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

1.2. Features

➢ Various CPU and OS for Choosing

MiniOS7
80186 CPU
µPAC-5000 Series

- DOS-like
- Boot up in 0.4 ~ 0.8 second
- Build-in hardware diagnostic
- Standard version for C language programming
- ISaGRAF version for IEC 61131-3 programming

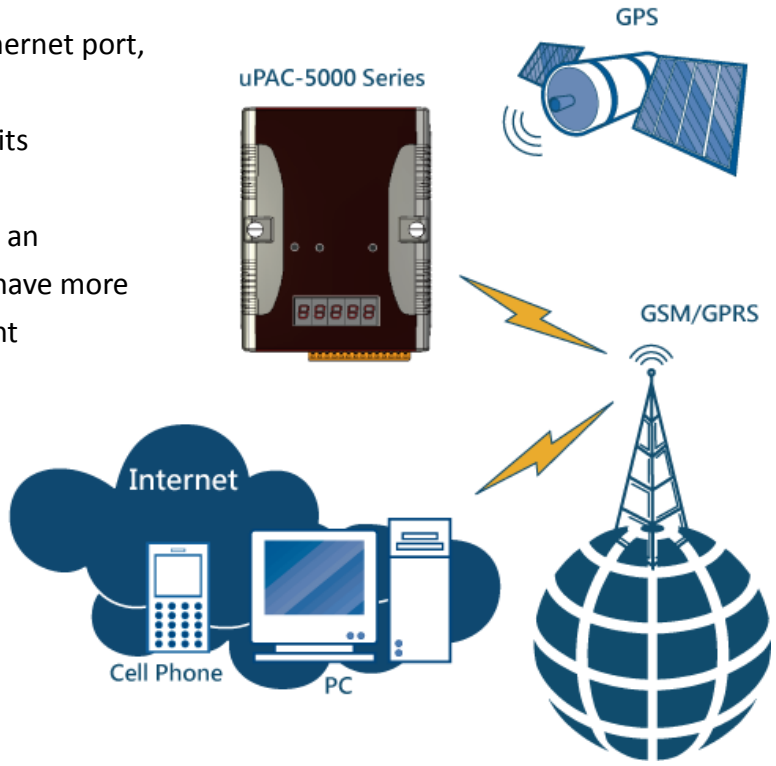
➢ Local I/O and Communication Expansion Board

The 5000 series equip an I/O expansion bus to support one optional expansion board, called XW-board. It can be used to implement various I/O functions such as DI, DO, A/D, D/A, Timer/Counter and various communication interface, such as RS-232/422/485, CAN, FRnet, etc.

Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

➢ Remote I/O Module and Expansion Unit

With the built-in RS-485 and Ethernet port, the 5000 series can connect RS-485/Ethernet remote I/O Units (RU-87Pn/ET-87Pn) or modules (I-7000/M-7000/ET-7000). With an XW-board, the 5000 series can have more communication ports or different interface to connect to other type of devices, for example: CANOpen devices, DeviceNet devices, or FRnet I/O modules.



➢ Multi-Communication Interface

There are several communication interfaces to expand I/O and connect external devices:

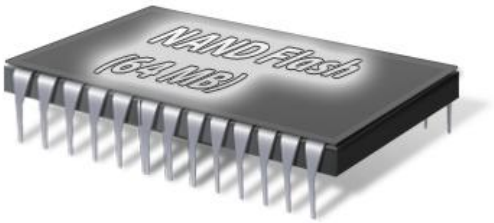
- Ethernet
- RS-232/422/485
- USB host
- CAN bus
- FRnet
- GSM/GPRS
- GPS

Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

➢ Various Memory Expansions

µPAC-5000 provides various memory storage options, such as EEPROM, Flash, battery-backup SRAM or microSD. Customers can choose the memory based on their characteristics.

- 16 KB EEPROM: to store not frequently changed parameters.
- microSD: to implement portable data logging applications.
  max. of 2GB on the MiniOS7 platform
  max. of 32 GB on Linux and WinCE platforms



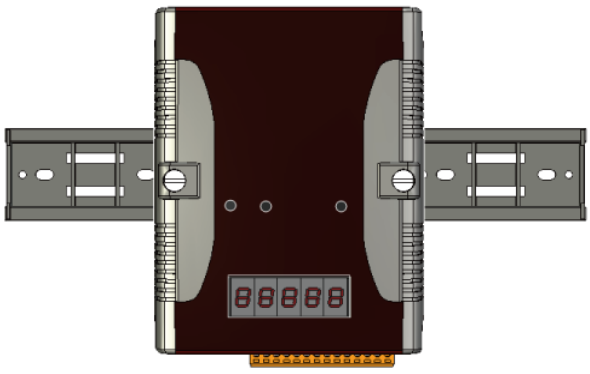
- 64 MB Flash: to implement non-portable data logging applications.
- 512 KB battery backup SRAM: to hold data while power lost.

➢ Unique 64-bit Hardware Serial Number to Protect Your Program

A unique 64-bit serial number is assigned to each hardware device to protect your software against piracy.

➢ Small and Easy Installation

µPAC -5000 series have a slender shape (91 mm x 132 mm x 52 mm) to be install in a narrow space with DIN-Rail.



Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

➤ Plastic and Metal Housing

The default housing is plastic material, customers can choose metal housing if they want.

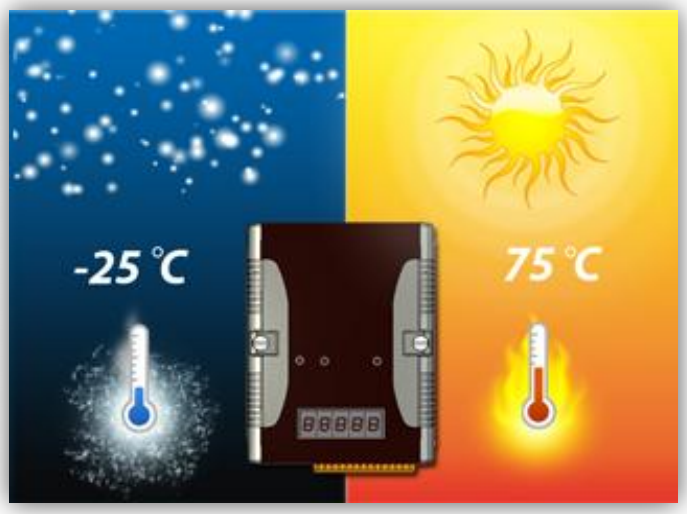
➤ Redundant Power Inputs

To prevent the µPAC-5000 from failing by the power loss, the power module is designed with two input connectors. Once a power input fails, the power module switches to the other power input. And there is a relay output for informing the power failure.

➤ Highly Reliable Under Harsh Environment

The µPAC-5000 operates in a wide range of temperature and humidity.

- Operating Temperature:
  -25°C ~ +75 °C
- Storage Temperature:
  -40°C ~ +80 °C
- Humidity:
  10 ~ 95% RH, non-condensing



Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

1.3. Specifications

1.3.1. General Series

Models	Regular Version	µPAC5001(D)	µPAC5001-FD(D)	µPAC5002(D)	µPAC5002-FD(D)	µPAC5002-NV(D)	µPAC5002-SM(D)
	PoE Version	µPAC5001P(D)	µPAC5001P-FD(D)	µPAC5002P(D)	µPAC5002P-FD(D)	µPAC5002P-NV(D)	µPAC5002P-SM(D)
System Software							
OS		MiniOS7 (DOS-like embedded operating system)					
Program Upload Interface		RS-232 (COM1) or Ethernet					
Programming Language		C language					
Compilers to Create .exe Files		TC++ 1.01/TC2.01 (Freeware)/BC++3.1 ~ 5.2x/MSC 6.0/MSVC++ (before version 1.5.2)					
CPU Module							
CPU		80186 or compatible (16-bit and 80MHz)					
SRAM		512 KB		768 KB			
Flash		512 KB; erase unit is on sector (64 K bytes); 100,000 erase/write cycles					
microSD Expansion		Yes, can support 1 or 2 GB microSD					
64 MB NAND Flash Disk		-	Yes	-	Yes	-	
1 MB NVRAM		-				Yes	-
512 KB Battery Backup SRAM		-					Yes
EEPROM		16 KB					
NVRAM		31 Bytes (battery backup, data valid up to 5 year)					
RTC (Real Time Clock)		RTC provide seconds, minutes, hours, date of week/month; month and year, valid from 1980 to 2079					
64-bit Hardware Serial Number		Yes					
Watchdog Timers		Yes (0.8 second)					
Communication Ports							
Ethernet		RJ-45 x 1, 10/100 Base-TX (Auto-negotiating, Auto MDI/MDI-X, LED indicators)					
COM1		RS-232 (TxD, RxD, RTS, CTS, GND), non-isolated, speed: 115200 bps max.					
COM2		RS-485 (D2+, D2-), self-tuner ASIC inside, non-isolated, speed: 115200 bps max.					
LED Indicator							
Programmable LED Indicators		2					
LED Display		5-digit 7-segment LED display for (D) versions					
Hardware Expansion							
I/O Expansion Bus		Yes					
Mechanical							
Dimension (W x H x D)		91 mm x 123 mm x 52 mm					
Installation		DIN-Rail					
Environmental							
Operating Temperature		-25 ~ + 75 ˚C					
Storage Temperature		-30 ~ +80 ˚C					
Ambient Relative Humidity		10 ~ 90 % RH (non-condensing)					
Power							
Protection		Power reverse polarity protection					
Frame Ground		Yes (for ESD protection)					
Input Range		+12 ~ +48 V_{DC}					
Isolation		-					
Redundant Power Inputs		Yes					
Power over Ethernet (PoE)	Regular version	-					
	PoE version	IEEE 802.3af Class 1					
Power Consumption		2 W; 2.5 W for (D) version					

Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

1.3.2. GPS Series

GPS Series coming soon.
Please visit our products page or contact us for more information.
http://www.icpdas.com/products/PAC/up-5000/selection_guide.htm

1.3.3. GPRS Series

GPRS Series coming soon.
Please visit our products page or contact us for more information.
http://www.icpdas.com/products/PAC/up-5000/selection_guide.htm

1.3.4. GPS & GPRS Series

GPS & GPRS Series coming soon.
Please visit our products page or contact us for more information.
http://www.icpdas.com/products/PAC/up-5000/selection_guide.htm

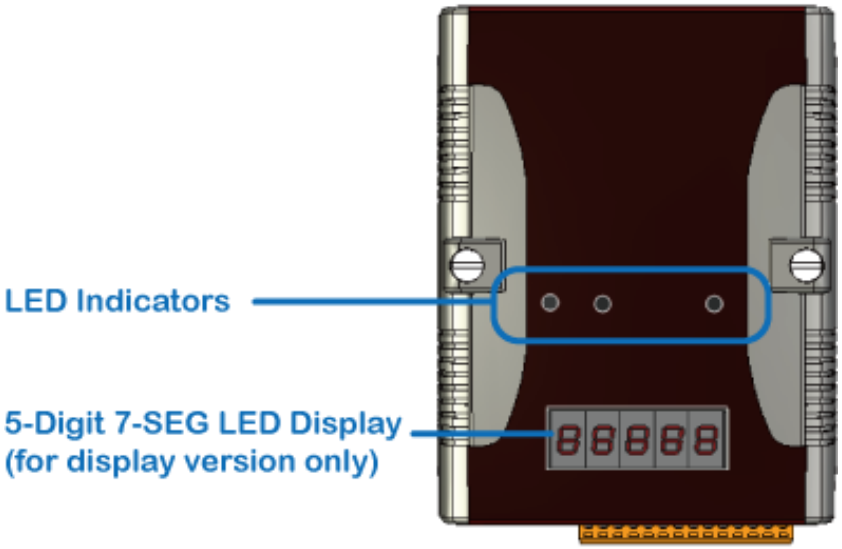
Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

1.4. Overview

Here is a brief overview of the components and its descriptions for module status.

Front Panel

The LED indicators and 5-digit 7-Segnment LED display are on the front panel that provides a very convenient way of displaying information for faster, easier diagnostics.



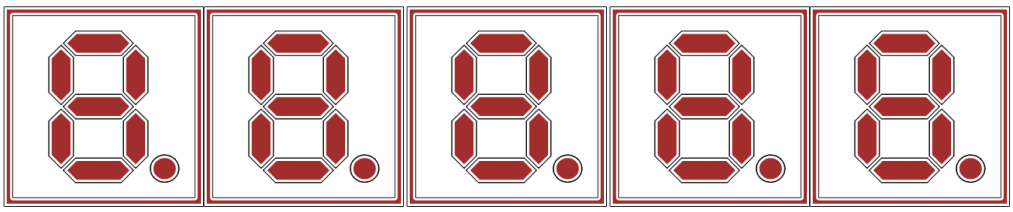
➤ LED Indicators

LED indicators are on the front panel of the µPAC-5000, their functions are summarized in the table below.

Indicator	State	Meaning
L1	Flashing	User programmable LED
L2	OFF	User programmable LED
Link (G)	ON	Ethernet link detected
	OFF	No Ethernet link detected
	Flashing Green	Ethernet packet received

Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

➤ 5-Digit 7-SEG LED display (for display version only)



µPAC-5000 display series equip 5-digital 7SEG LED display that can be used to display decimal numbers from 0 to 9 and provided a very convenient way of displaying digital data in the form of Numbers

Top Panel

The microSD memory socket is on the top panel that provides a simple way of expanding capacity.

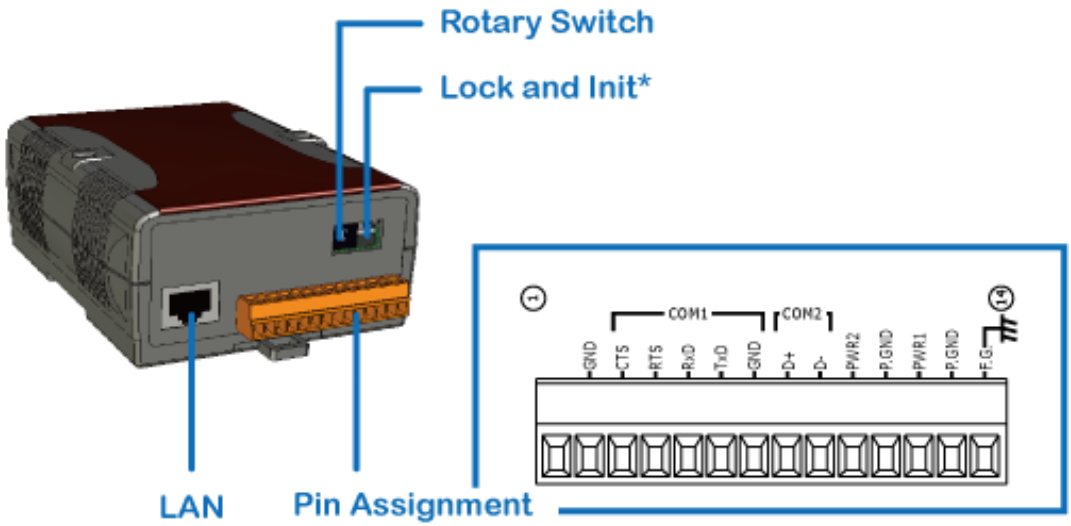
➤ microSD Memory Socket

µPAC-5000 equip a microSD socket and it can support up to 2 GB microSD card.

Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

Bottom Panel

The switches and interface are on the bottom panel that provides a simple way of adjusting the system and wiring the connection.



➤ Init Switch: Operating Mode Selector Switch

ON: MiniOS7 configuration mode
OFF: Firmware running mode

In the µPAC-5000 series, the Switch is always in the OFF position. Only when updating the µPAC-5000 firmware or OS, the switch can be moved from the OFF position to the ON position.

Move the Switch to the OFF position after the update is complete.

➤ Lock Switch: Flash Memory Write Protection Switch

ON: Enable Write Protection
OFF: Disable Write Protection

µPAC-5000 Flash memory with Write Protection can physically lock that prevents modification or erasure of valuable data on µPAC-5000.

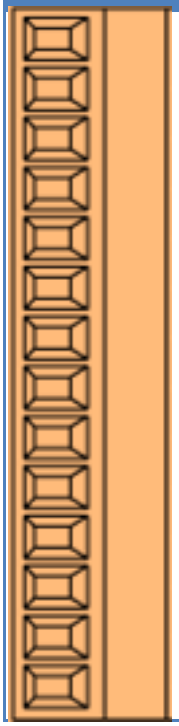
Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

➤ LAN

An Ethernet port is an opening on µPAC-5000 network equipment that Ethernet cables plug into. Ethernet ports accept cables with RJ-45 connectors.

➤ Pin Assignment

The pin assignments of the connector are as follows:



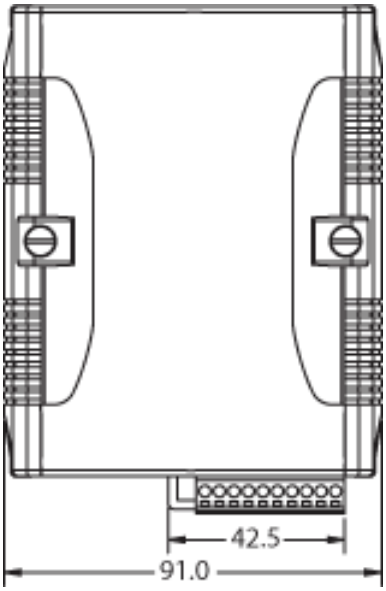
Pin	Signal	Description
1	N.C	Unassigned
2	GND	Ground
3	CTS	RS-232
4	RTS	
5	RxD	
6	TxD	
7	GND	
8	D+	RS-485
9	D-	
10	PWR2	Power Input 1
11	P.GND	
12	PWR1	Power Input 2
13	P.GND	
14	F.G.	Frame Ground

Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

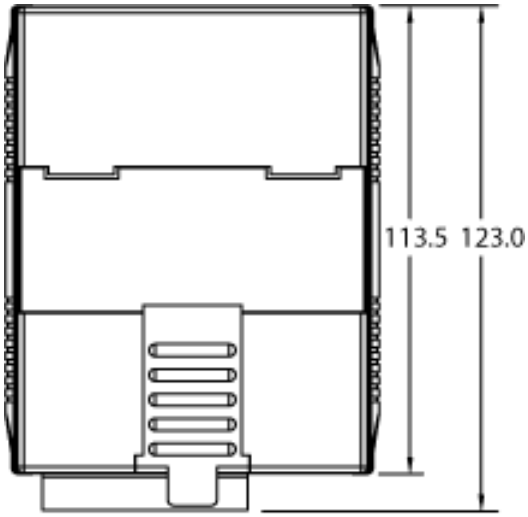
1.5. Dimensions

All dimensions are in millimeters.

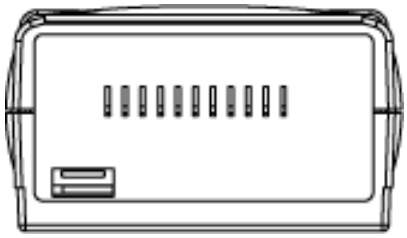
Front View



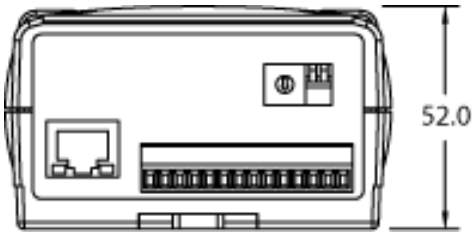
Back View



Top View



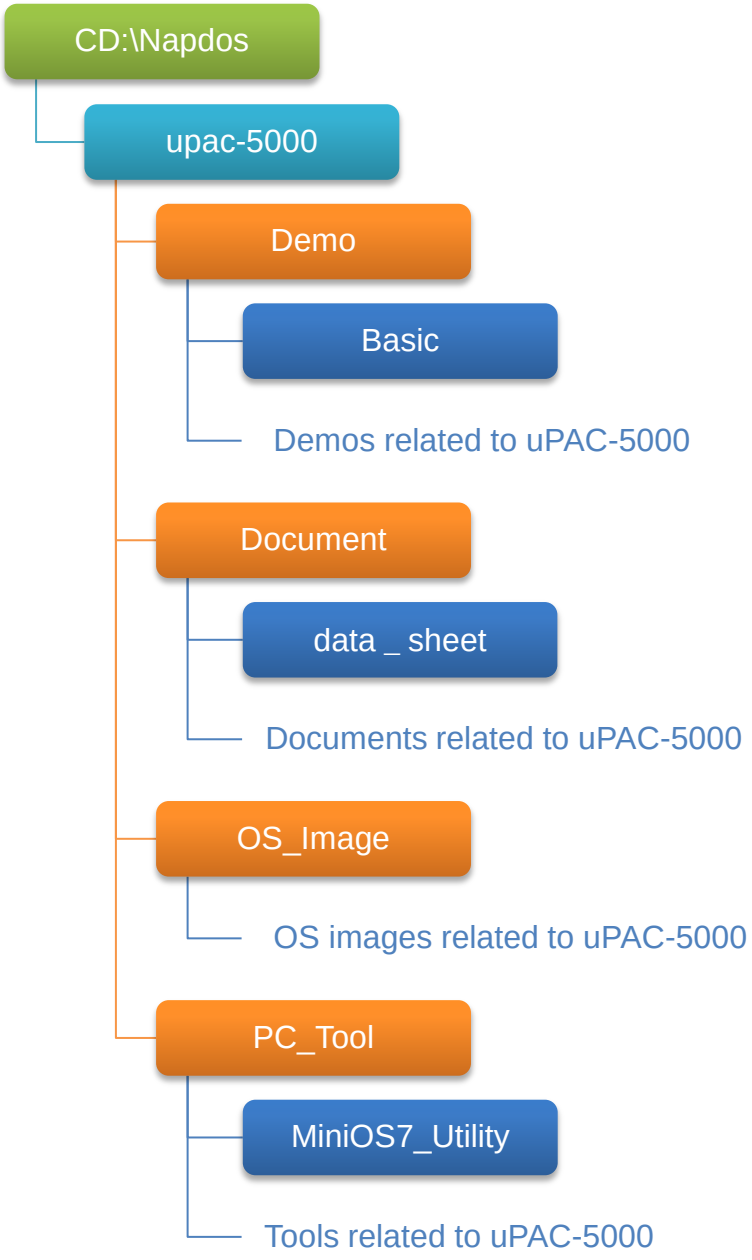
Bottom View



Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

1.6. Companion CD

This package comes with a CD that provides drivers, software utility, all of the required documentations…, etc. All of them are listed below.



Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

2. Getting Started

If you are a new user, begin with this chapter, it includes a guided tour that provides a basic overview of installing, configuring and using the µPAC-5000.

Before beginning any installation, please check the package contents. If any items are damaged or missing, please contact us.

In addition to Quick Start Guide, the package includes the following items:

µPAC-5000

Software Utility CD

RS-232 cable (CA-0910)

Screw Driver (1C016)

Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

2.1. Hardware Installation

Before installing the hardware, you should have a basic understanding of hardware specification, such as the size of hard drive, the usable input-voltage range of the power supply, and the type of communication interfaces.
For complete hardware details, please refer to section “1.2. Specifications”

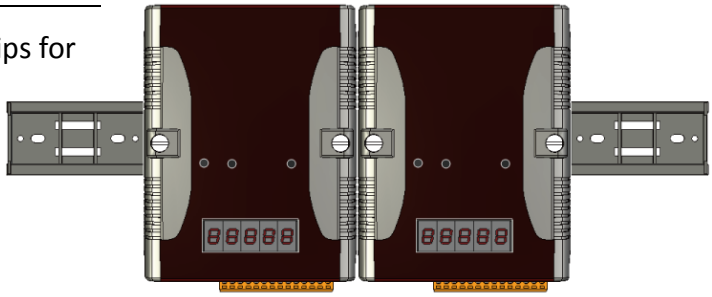
Below are step-by-step instructions for deploying the basic µPAC-5000 system.

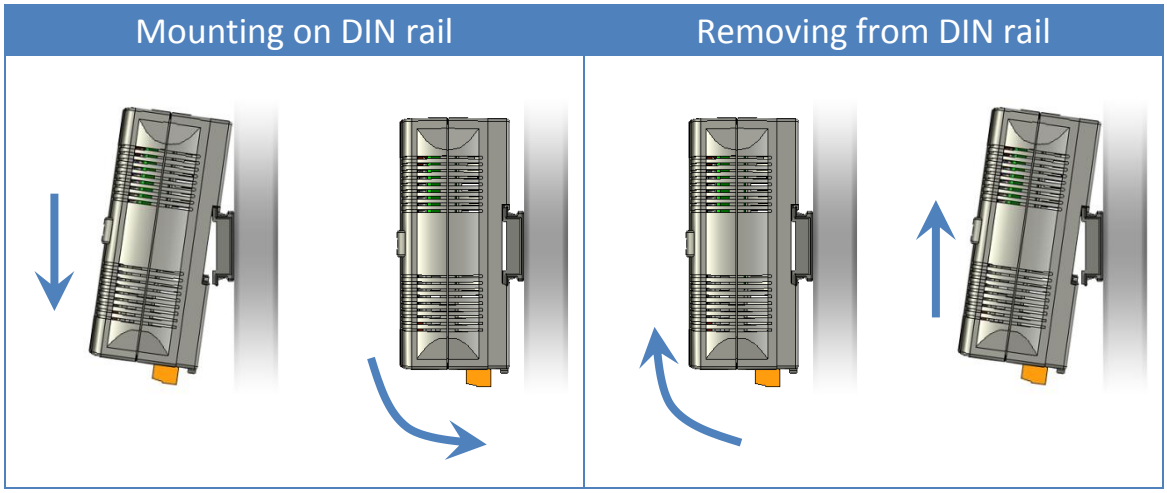
Step 1: Mount the hardware

The µPAC-5000 can be mounted with the bottom of the chassis on the DIN rail or piggyback.

➤ **DIN Rail mounting**

The µPAC-5000 has simple rail clips for mounting reliably on a standard 35 mm DIN rail.

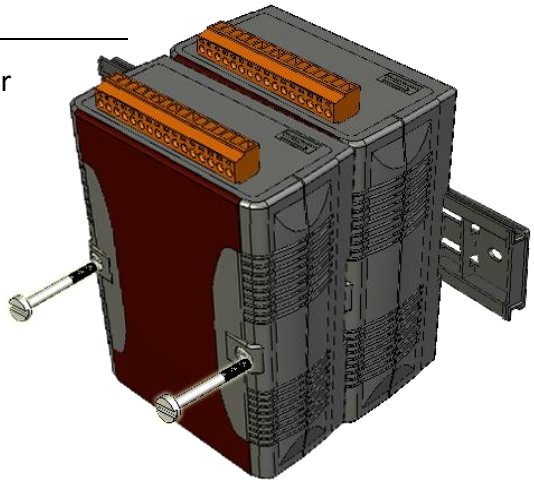




Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

➢ Piggyback mounting

The µPAC-5000 has two holes on both sides for piggyback mounting.



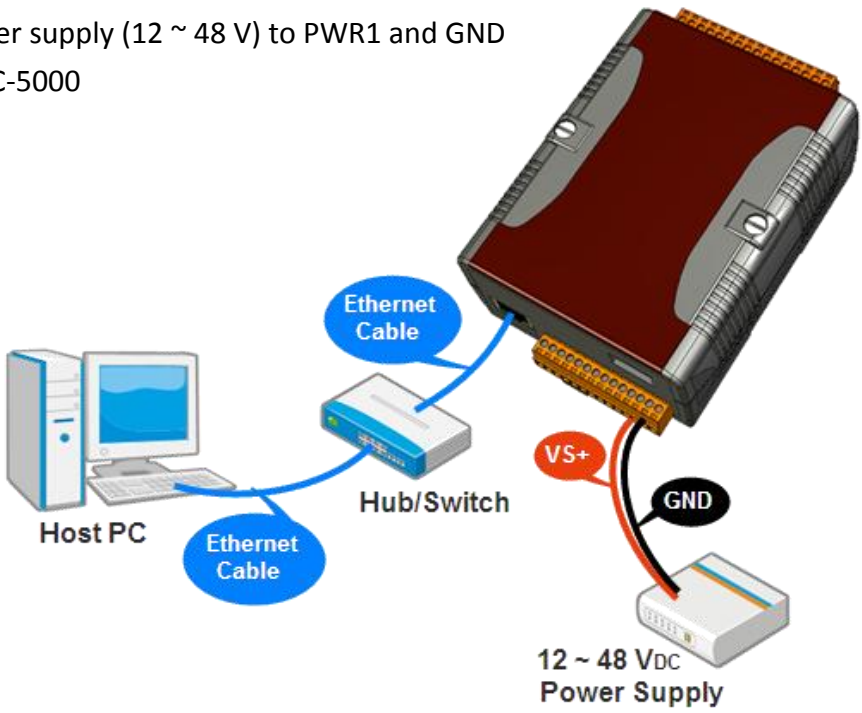
Step 2: Connect the µPAC-5000 to PC and setting up the power supply

The µPAC-5000 equip an RJ-45 Ethernet port for connection to an Ethernet hub/switch and PC, and powered by a standard 12 V_{DC} power supply or Power-over-Ethernet (PoE) switch which powers all components.

➢ External power supply via a standard 12 V_{DC} power supply

i. Connect PC to LAN port of µPAC-5000.

ii. Connect the power supply (12 ~ 48 V) to PWR1 and GND terminals of µPAC-5000



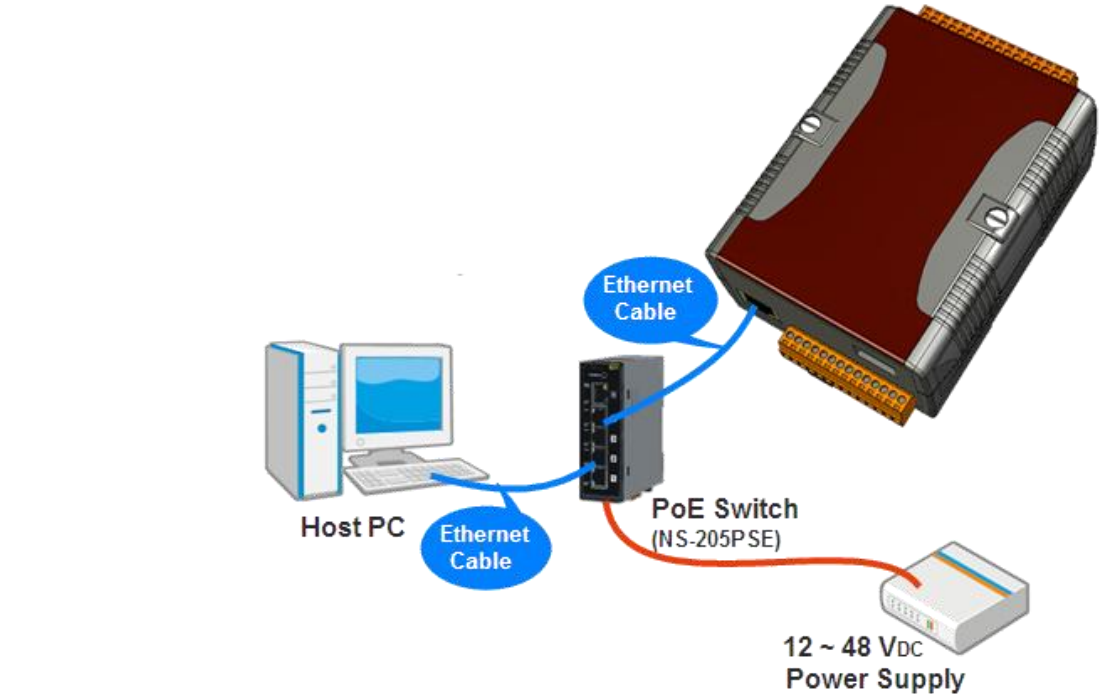
Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

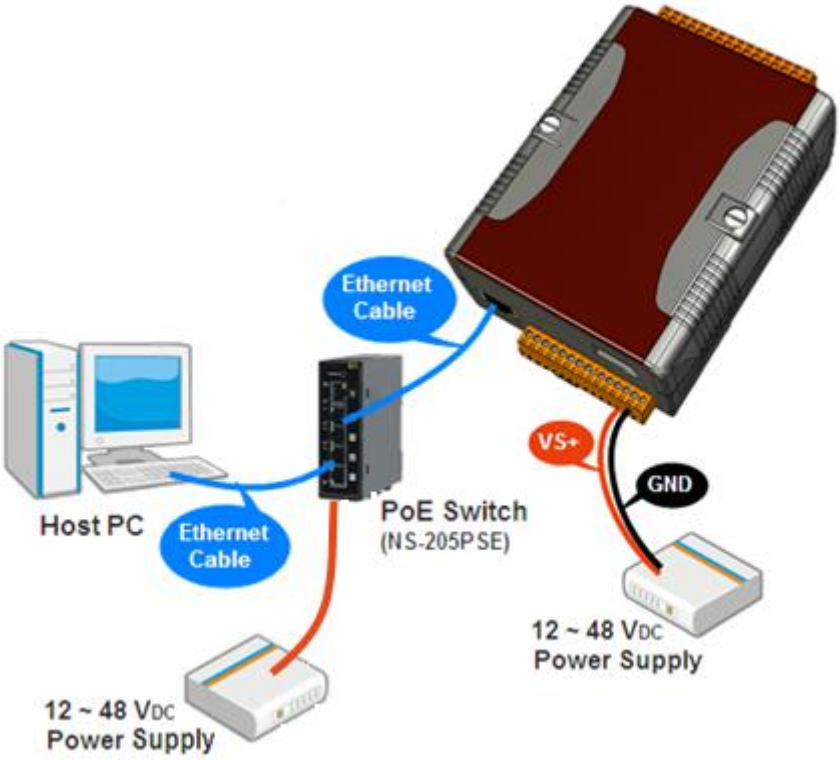
➤ External power supply via a PoE switch

i. Connect PC to the PoE switch

ii. Connect PoE switch to LAN port of µPAC-5000

iii. Connect the power supply (12 ~ 48 V_{DC}) to the PoE switch





Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

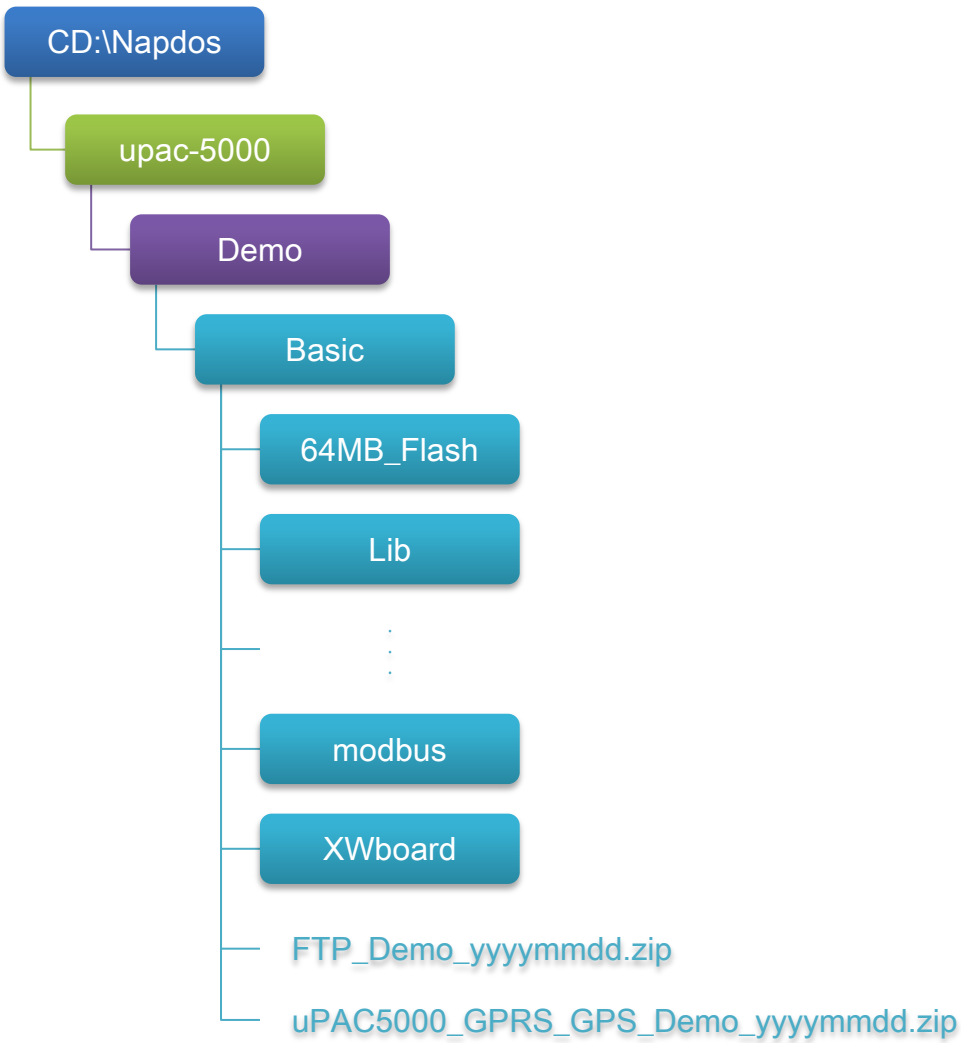
2.2. Software Installation

The Companion CD includes complete sets of APIs, demo programs and other tools for developing your own applications.

Below are step-by-step instructions for installing the µPAC-5000 APIs, demo programs and tools.

Step 1: Copy the “Demo” folder from the companion CD to PC

The folder is an essential resource for users developing your own applications which contains libraries, header files, demo programs and more information as shown below.



Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

Step 2: Installing the MiniOS7 Utility

MiniOS7 Utility is a suite of tool for managing MiniOS7 devices (µPAC-5000, iPAC-8000, µPAC-7186,. etc.). It's comprised of four components – System monitor, communication manager, file manager and OS loader.

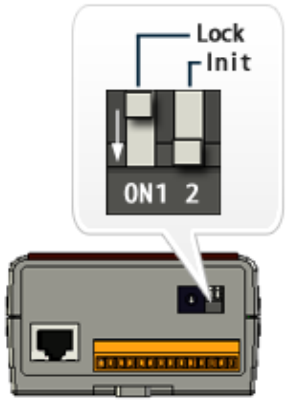
The MiniOS7 Utility can be obtained from companion CD or our FTP site:
CD:\Napdos\minios7\utility\minios7_utility\
ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/minios7/utility/minios7_utility/

Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

2.3. Boot Configuration

Before you upload some programs to µPAC-5000, you need to enter the Init mode and disable the Write Protection.

Make sure the switch of the Lock placed in the “OFF” position, and the switch of the Init placed in the “ON” position.



Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

2.4. Uploading µPAC-5000 Programs

MiniOS7 Utility is a suite of tool for managing MiniOS7 devices (µPAC-5000, iPAC-8000, µPAC-7186,. etc.). It's comprised of four components – System monitor, communication manager, file manager and OS loader.

Before you begin using the MiniOS7 Utility to upload programs, ensure that µPAC-5000 is connected to PC.

The upload process has the following main steps:

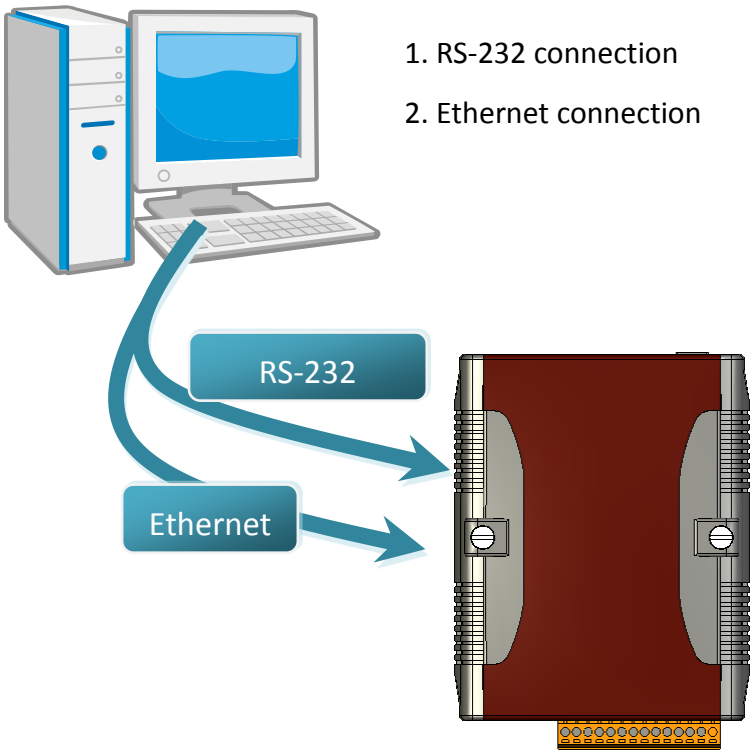
1. Establishing a connection between PC and µPAC-5000
2. Uploading and executing programs on µPAC-5000
3. Making programs start automatically

All of these main steps will be described in detail later.

Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

2.4.1. Establishing a connection between PC and µPAC-5000

There are two ways to establish a connection between PC and µPAC-5000.



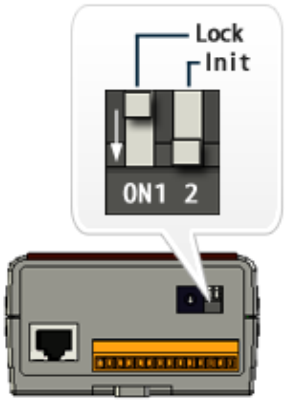
Each of the connection types will be described in detail later.

Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

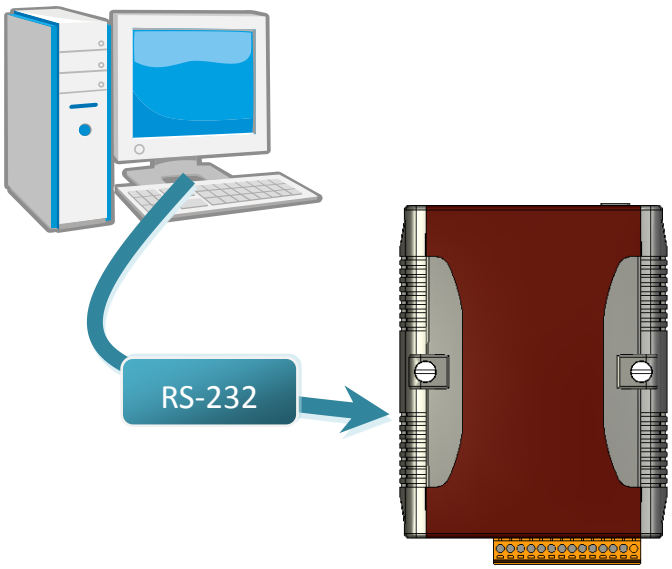
2.4.1.1. RS-232 connection

Below are step-by-step instructions on how to connect to PC using a RS-232 connection.

Step 1: Turn the switch of the Lock to “OFF” position, and the switch of Init to “ON” position



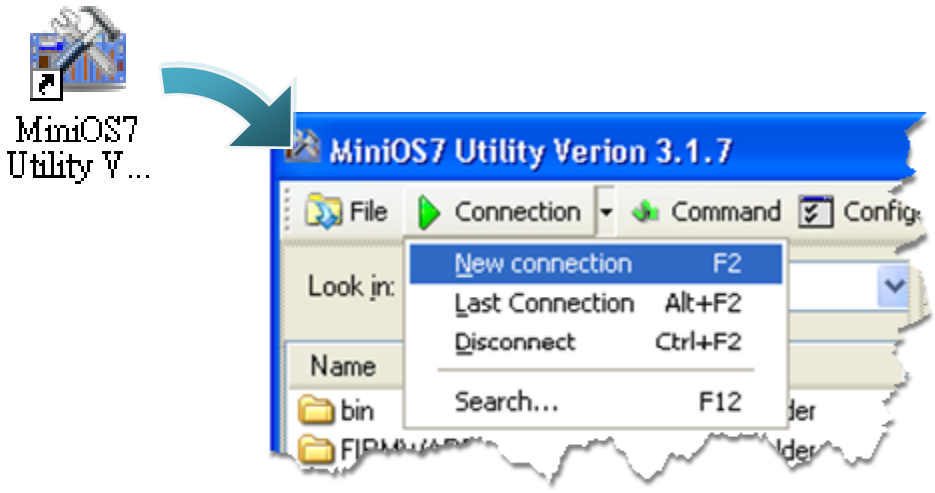
Step 2: Use the RS-232 Cable (CA-0910) to connect to PC



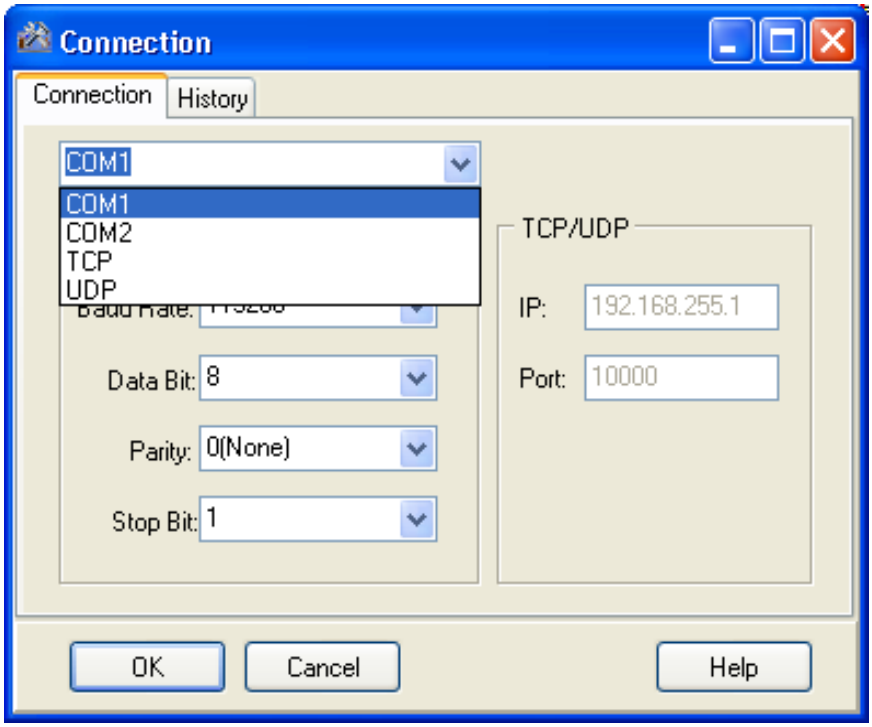
Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

Step 3: Run the MiniOS7 Utility

Step 4: Click the “New connection” function from the “Connection” menu

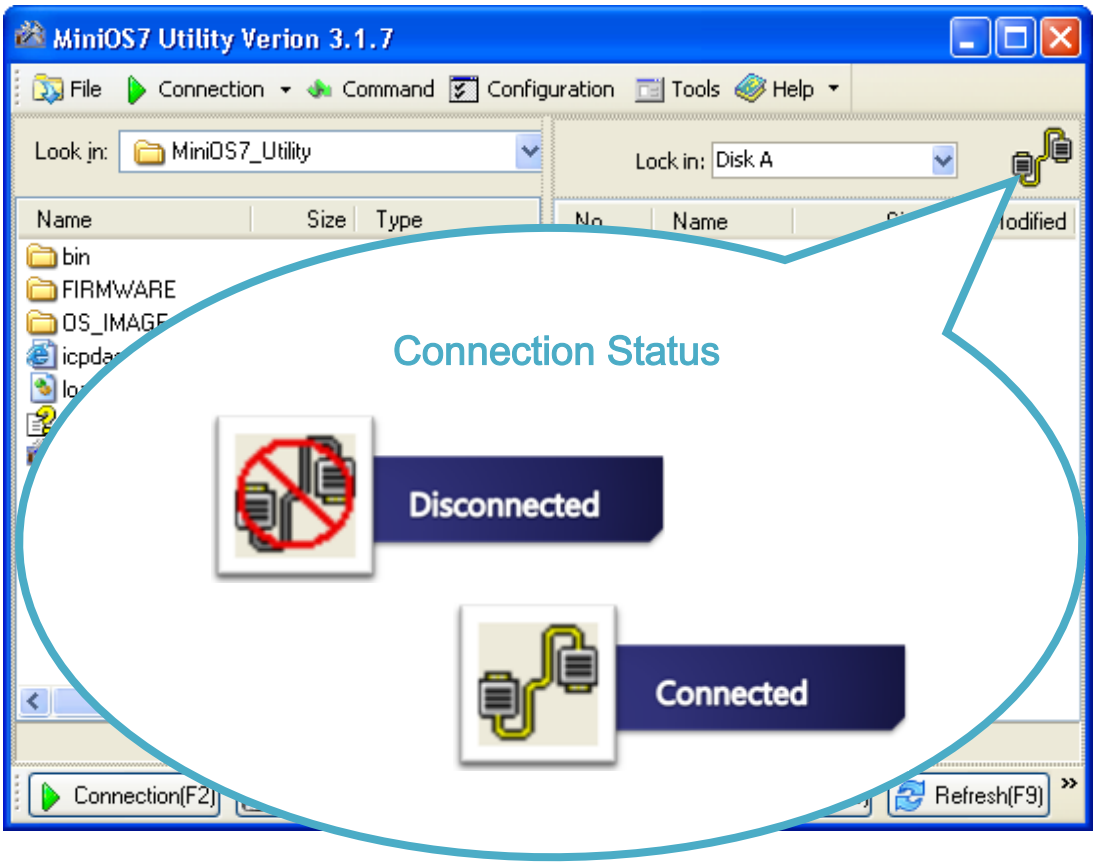


Step 5: On the “Connection” tab of the “Connection” dialog box, select “COM1” from the drop down list, and then click “OK”



Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

Step 6: The connection has already established

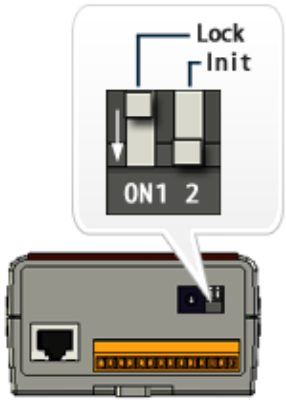


Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

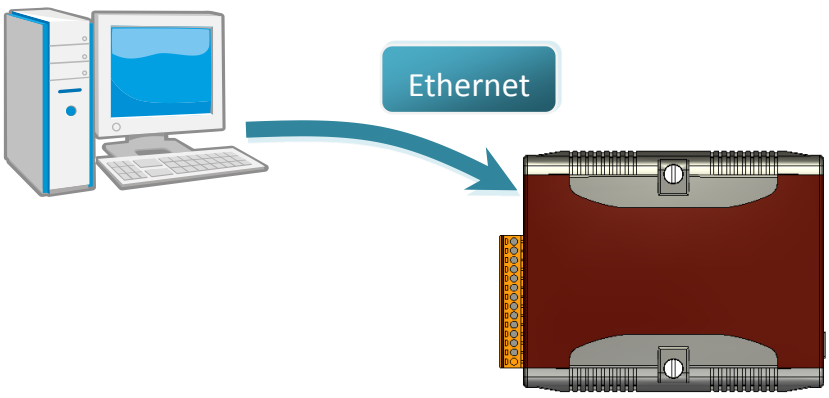
2.4.1.2. Ethernet Connection

Below are step-by-step instructions on how to connect to PC using a Ethernet connection.

Step 1: Turn the switch of the Lock to “OFF” position, and the switch of Init to “ON” position



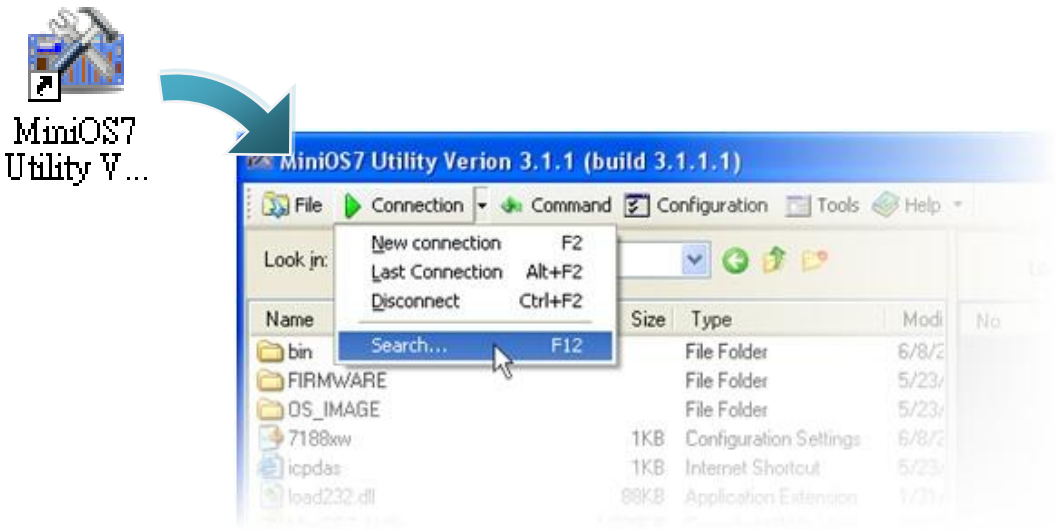
Step 2: Use an Ethernet cable to connect to PC



Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

Step 3: Run the MiniOS7 Utility

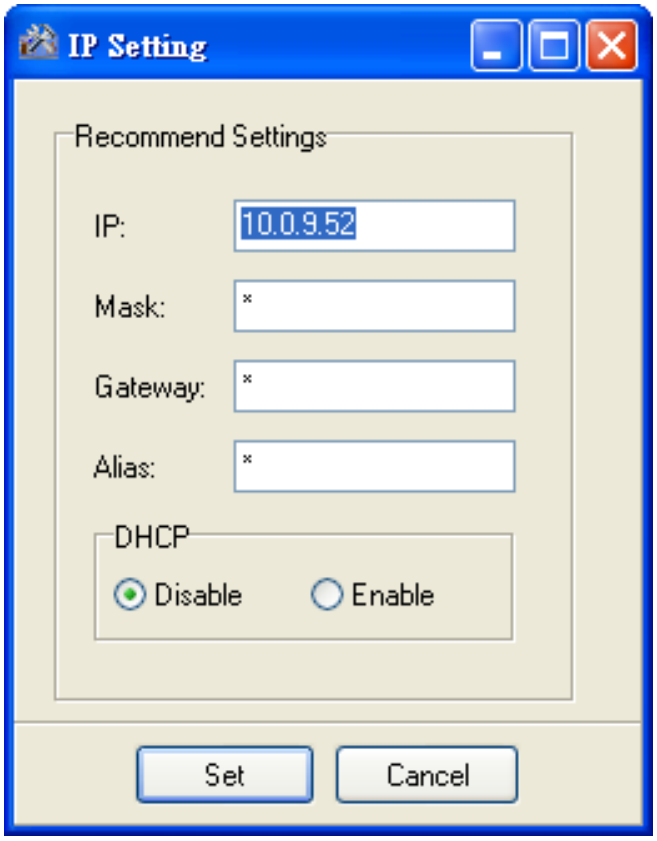
Step 4: Click the “Search” function from the “Connection” menu



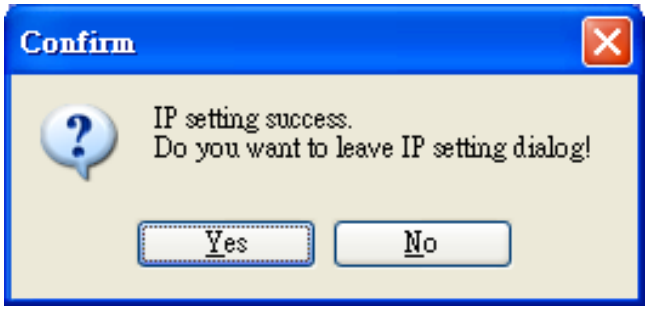
Step 5: On the “MiniOS7 Scan” dialog box, choose the module name from the list and then choose “IP setting” from the toolbar

Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

Step 6: On the “IP Setting” dialog, configure the “IP” settings and then click the “Set” button

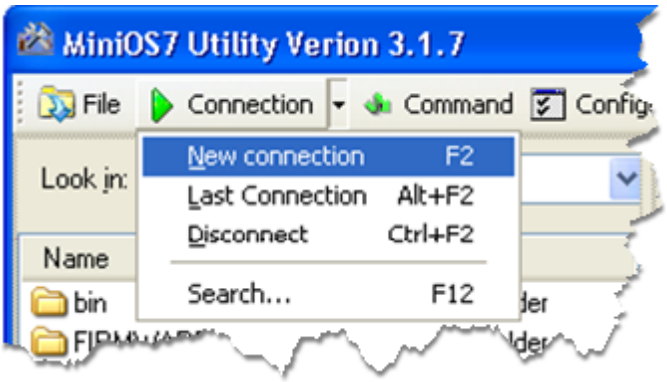


Step 7: On the “Confirm” dialog box, click “Yes”

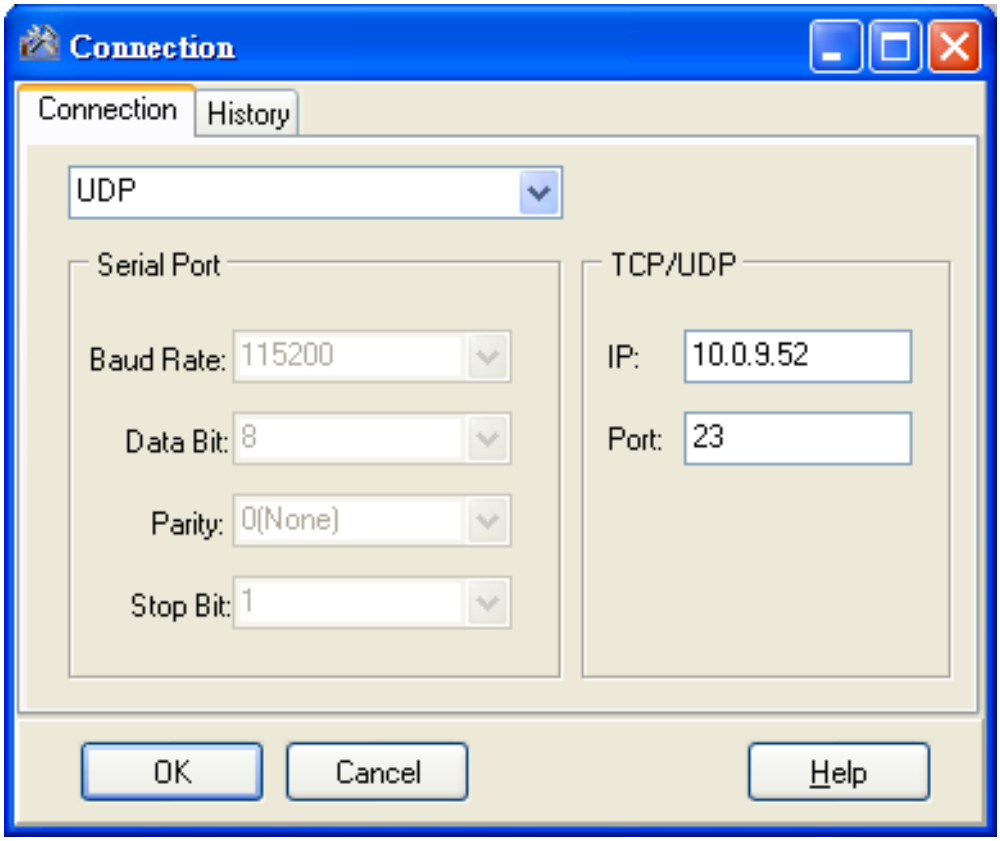


Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

Step 8: Click the “New connection” function from the “Connection” menu

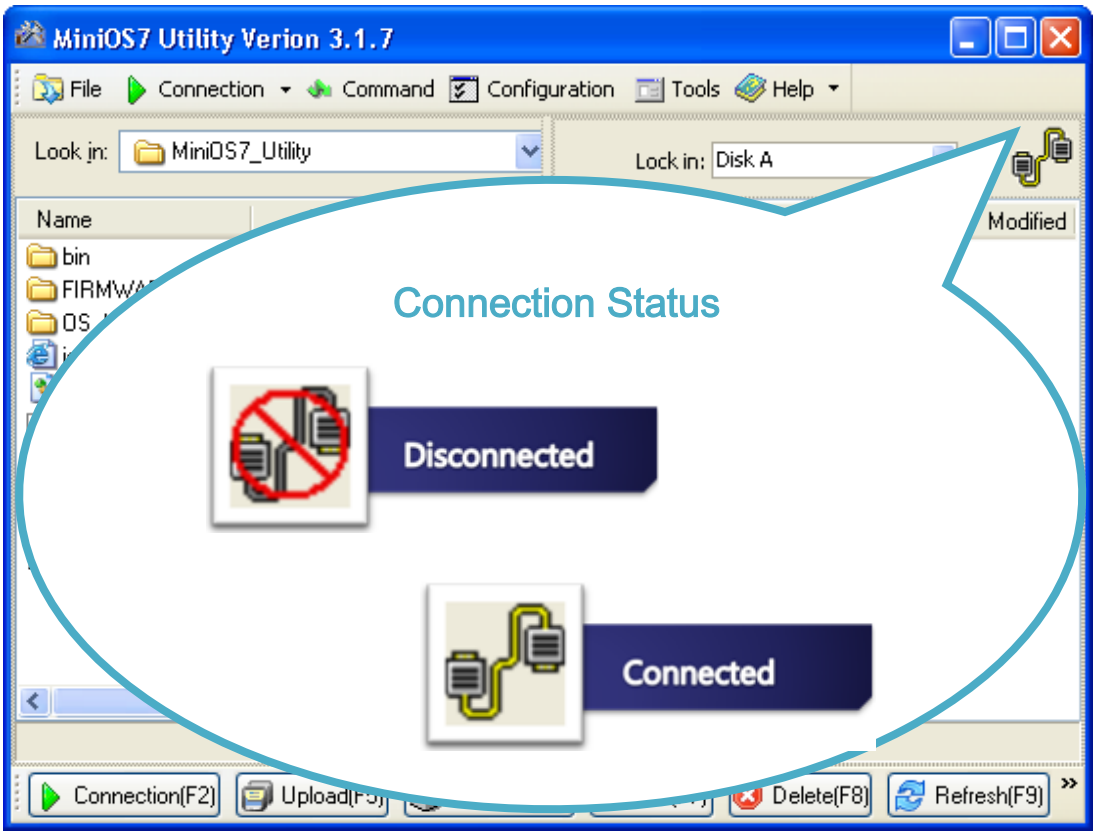


Step 9: On the “Connection” tab of the “Connection” dialog box, select “UDP” from the drop down list, type the IP address which you are assigned, and then click “OK”



Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

Step 10: The connection has already established

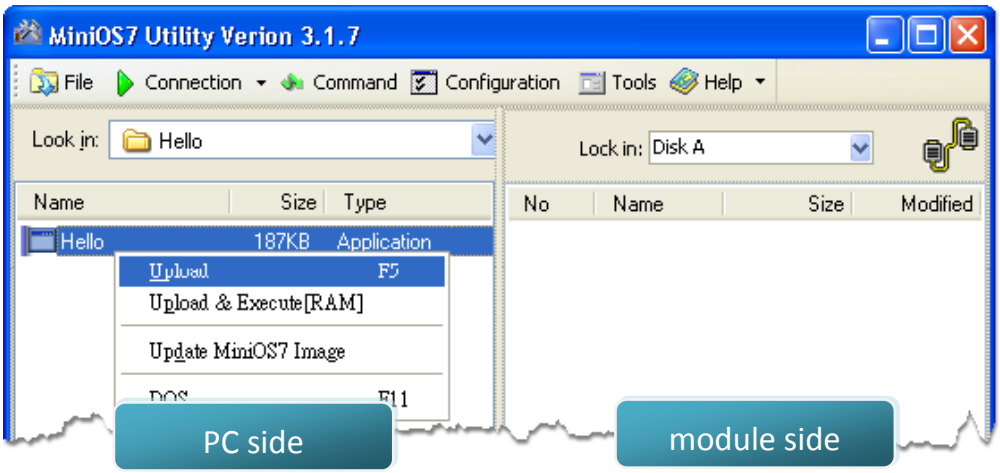


Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

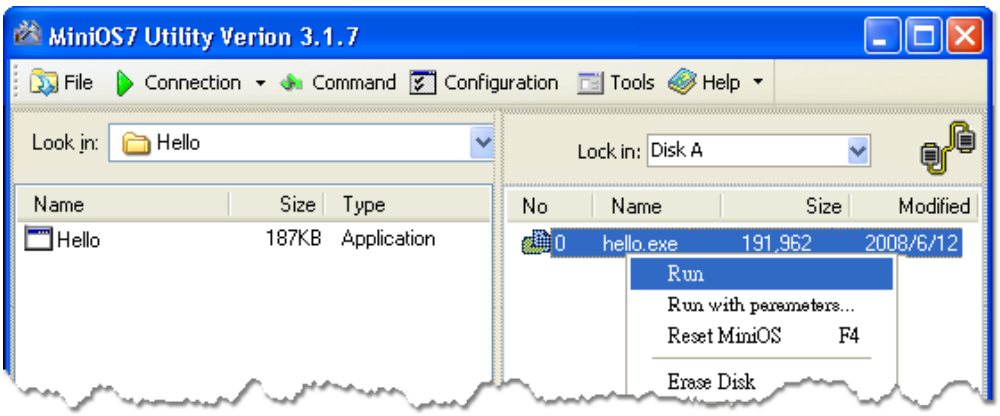
2.4.2. Uploading and executing µPAC-5000 programs

Before uploading and executing µPAC-5000 programs, you must firstly establish a connection between PC and µPAC-5000, for more detailed information about this process, please refer to section “2.4.1. Establishing a connection”

Step 1: On PC side, right click the file name that you wish to upload and then select the “Upload”



Step 2: On the module side, right click the file name that you wish to execute and then select the “Run”



Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

2.4.3. Making programs start automatically

After upload programs on the µPAC-5000, if you need programs to start automatically after the µPAC-5000 start-up, it is easy to achieve it, to create a batch file called autoexec.bat and then upload it to the µPAC-5000, the program will start automatically in the next start-up.

For example, to make the program “hello” run on start-up.

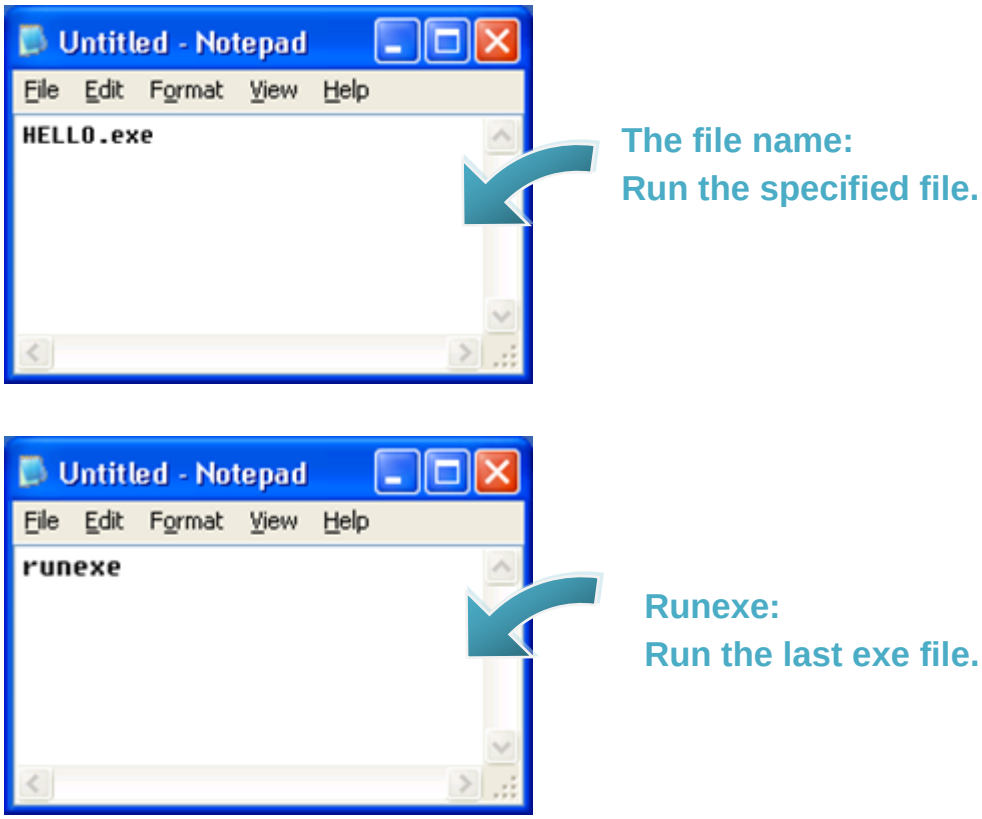
Step 1: Create an autoexec.bat file

i. Open the “Notepad”

ii. Type the command

The command can be either the file name “HELLO.exe” (run the specified file) or “runexe” (run the last exe file)

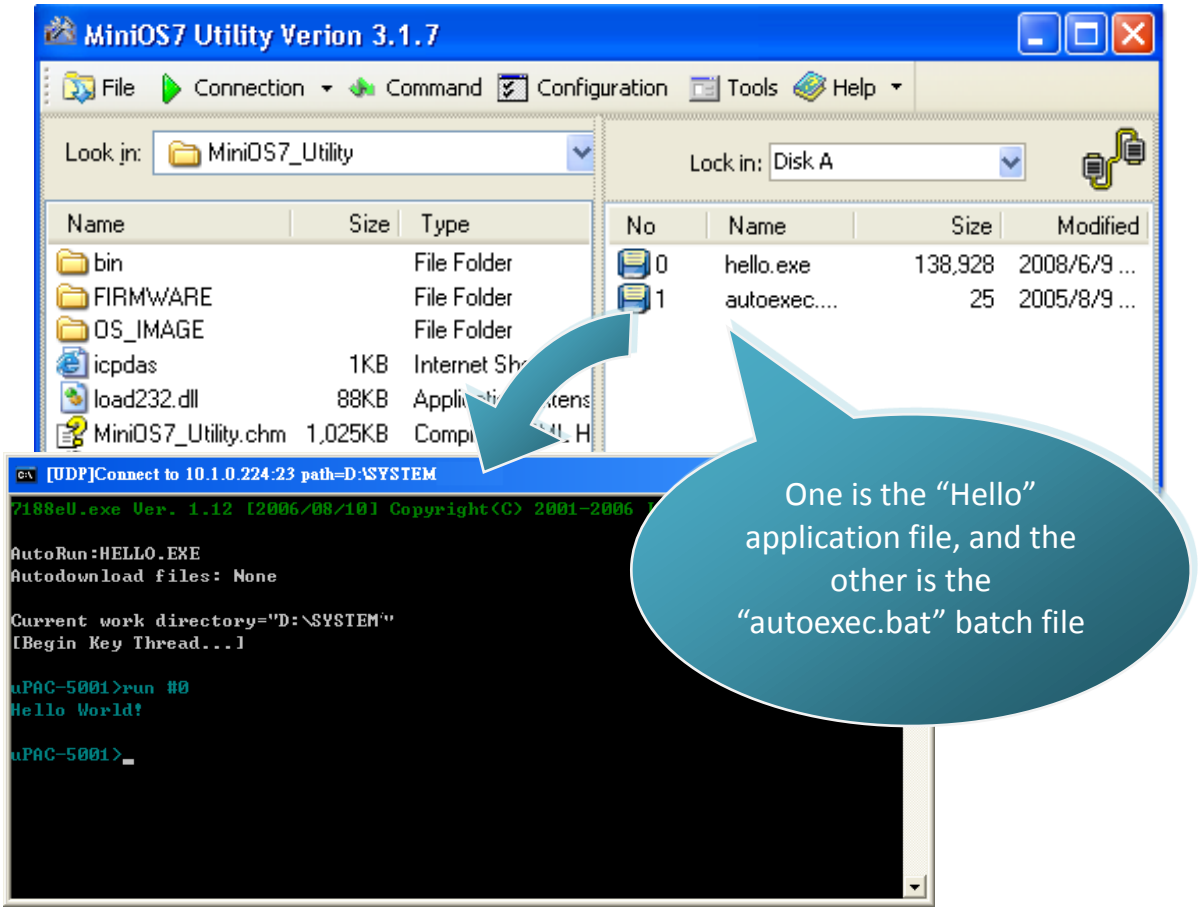
iii. Save the file as autoexec.bat



Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

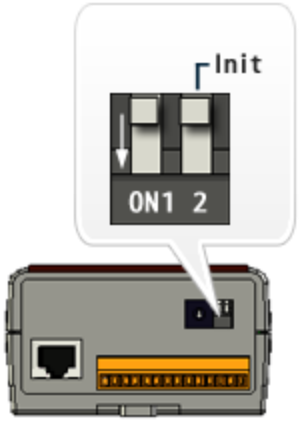
Step 2: Upload programs to µPAC-5000 using MiniOS7 Utility

For more detailed information about this process, please refer to section “2.4.1. Establishing a connection”



Tips & Warnings

Before restaring the module for settings to take effect, you must firstly turn the switch of Init to “OFF” position.



Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

2.5. Updating µPAC-5000 OS image

ICP DAS will continue to add additional features to µPAC-5000 in the future, we advise you periodically check the ICP DAS web site for the latest update to µPAC-5000.

Step 1: Get the latest version of the µPAC-5000 OS image

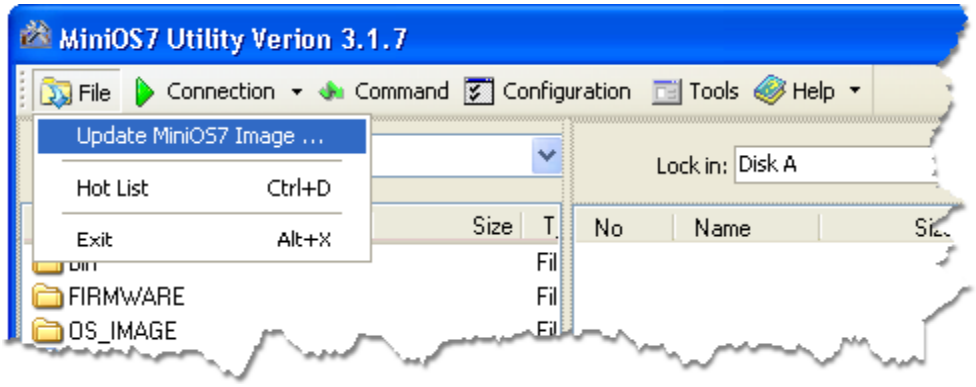
The latest version of the µPAC-5000 OS image can be obtained from:
CD:\NAPDOS\upac-5000\OS_image\
http://ftp.icpdas.com/pub/cd/8000cd/napdos/upac-5000/os_image/

Step 2: Establish a connection

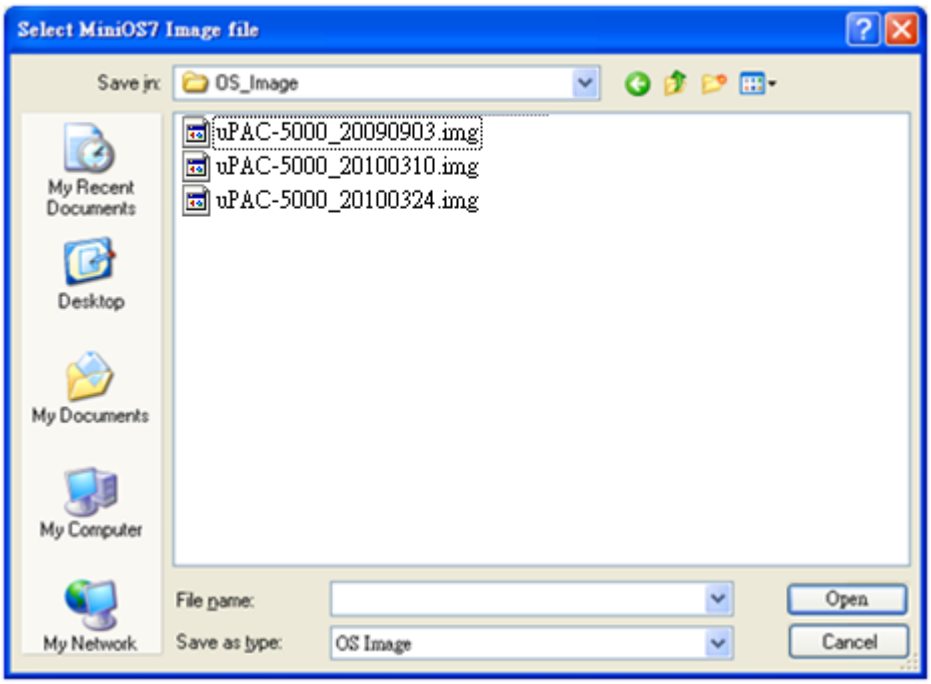
For more detailed information about this process, please refer to section “2.4.1. Establishing a connection”

Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

Step 3: Click the “Update MiniOS7 Image …” from the “File” menu

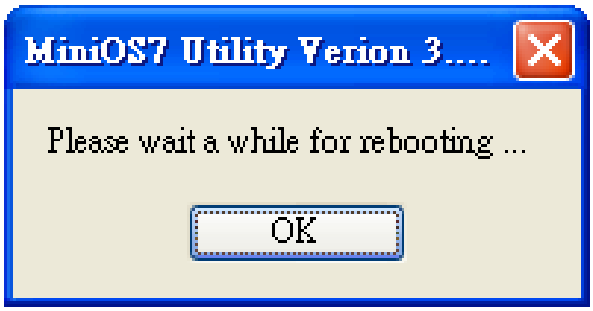


Step 4: Select the latest version of the MiniOS7 OS image

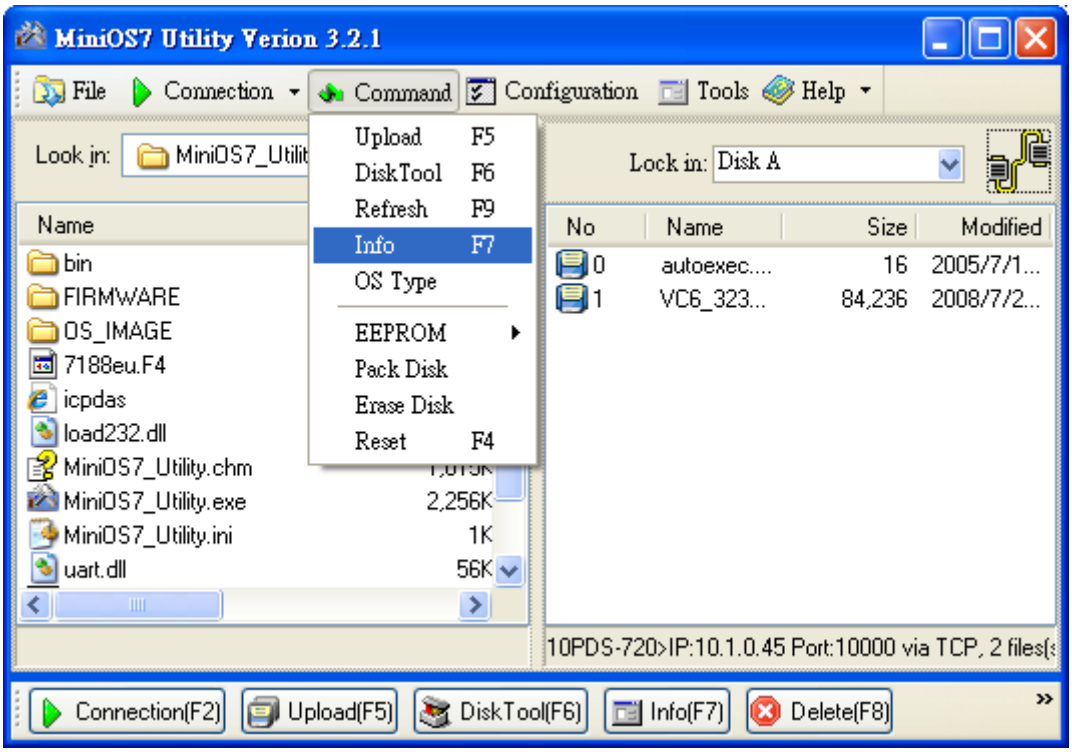


Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

Step 5: Click the “OK”



Step 6: Click the “Info” from the “Command” menu to check the version of the OS image



Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

3. “Hello World” - Your First Program

When you learn every computer programming language you may realize that the first program to demonstrate is "Hello World", it provides a cursory introduction to the language's syntax and output.

Below are step-by-step instructions on how to write your first µPAC-5000 program.

Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

3.1. C Compiler Installation

C is prized for its efficiency, and is the most popular programming language for writing applications.

Before writing your first µPAC-5000 program, ensure that you have the necessary C/C++ compiler and the corresponding functions library on your system.

The following is a list of the C compilers that are commonly used in the application development services.

- Turbo C++ Version 1.01
- Turbo C Version 2.01
- Borland C++ Versions 3.1 - 5.2.x
- MSC
- MSVC ++

We recommend that you use Borland C++ compiler as the libraries have been created on the companion CD.

Tips & Warnings

Before compiling an application, you need to take care of the following matters.

Generate a standard DOS executable program

Set the CPU option to 80188/80186

Set the floating point option to EMULATION if floating point computation is required. (Be sure not to choose 8087)

Cancel the Debug Information function as this helps to reduce program size. (MiniOS7 supports this feature.).

Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

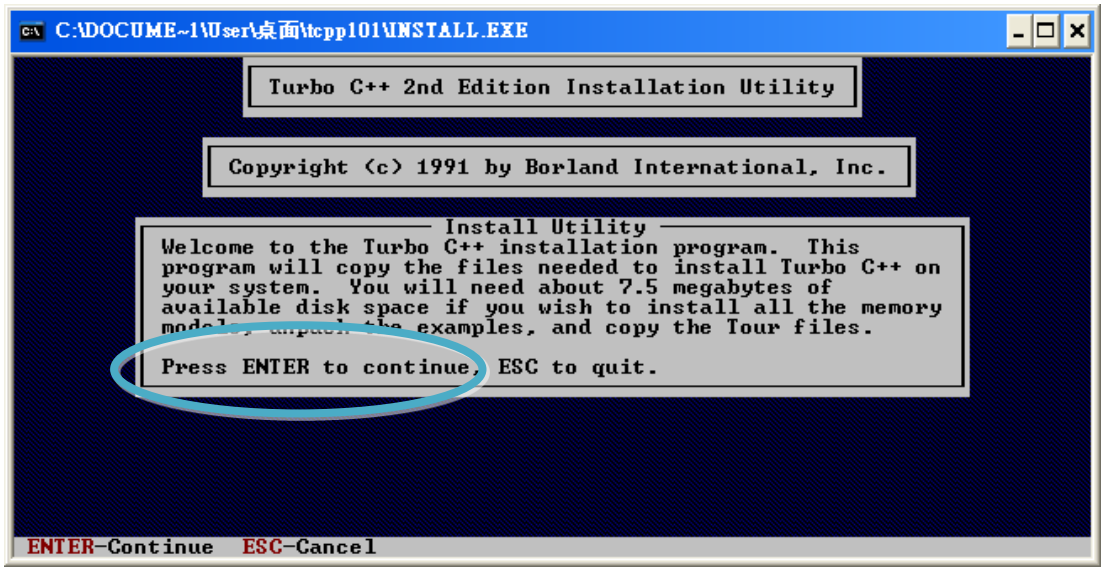
3.1.1. Installing the C compiler

If there is no compiler currently installed on your system, installation of the compiler should be the first step.

Below are step-by-step instructions for guiding you to install Turbo C++ Version 1.01 on your system.

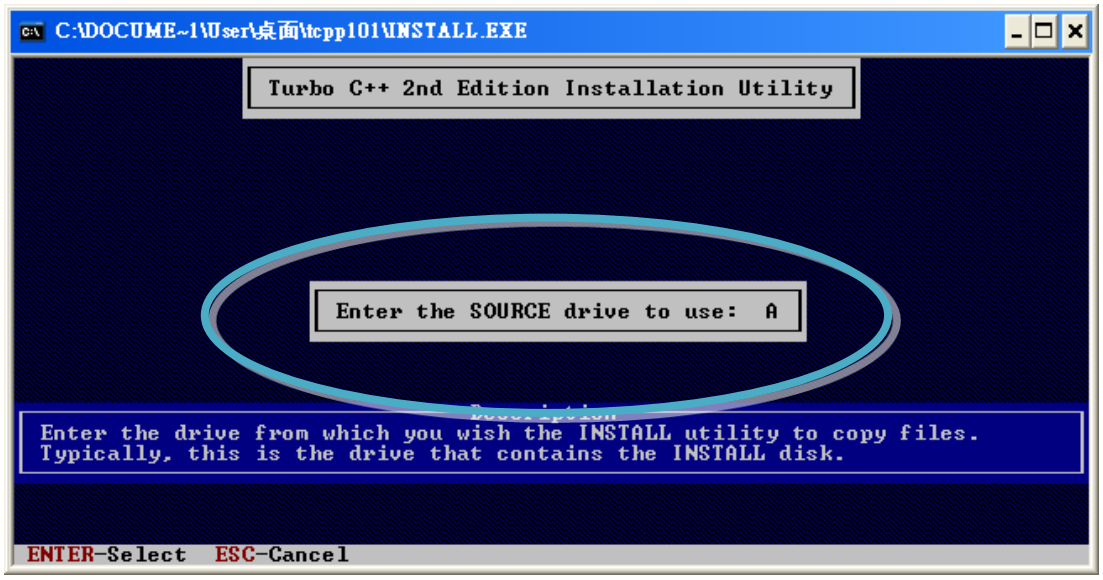
Step 1: Double click the Turbo C++ executable file to start setup wizard

Step 2: Press “Enter” to continue

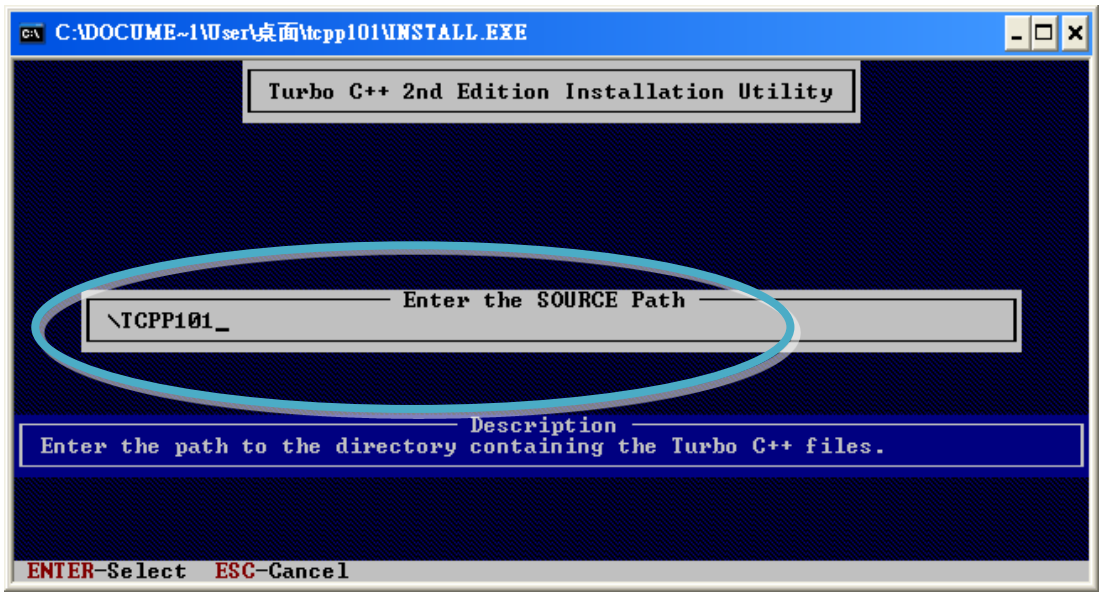

C:\DOCUME~1\User\桌面\tcpp101\INSTALL.EXE
Turbo C++ 2nd Edition Installation Utility
Copyright (c) 1991 by Borland International, Inc.
Install Utility
Welcome to the Turbo C++ installation program. This program will copy the files needed to install Turbo C++ on your system. You will need about 7.5 megabytes of available disk space if you wish to install all the memory
examples, and copy the Tour files.
Press ENTER to continue, ESC to quit.
ENTER-Continue ESC-Cancel

Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

Step 3: Enter the letter of the hard drive you wish to install the software

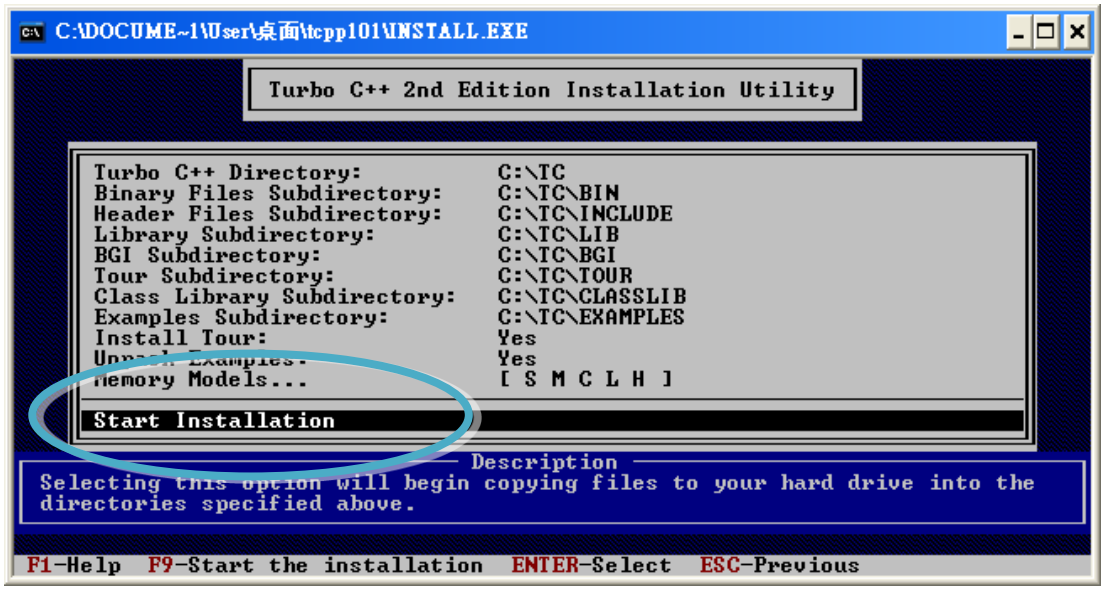

C:\DOCUME~1\User\桌面\tcpp101\INSTALL.EXE
Turbo C++ 2nd Edition Installation Utility
Enter the SOURCE drive to use: A
Enter the drive from which you wish the INSTALL utility to copy files.
Typically, this is the drive that contains the INSTALL disk.
ENTER-Select ESC-Cancel

Step 4: Enter the path to the directory you wish to install files to

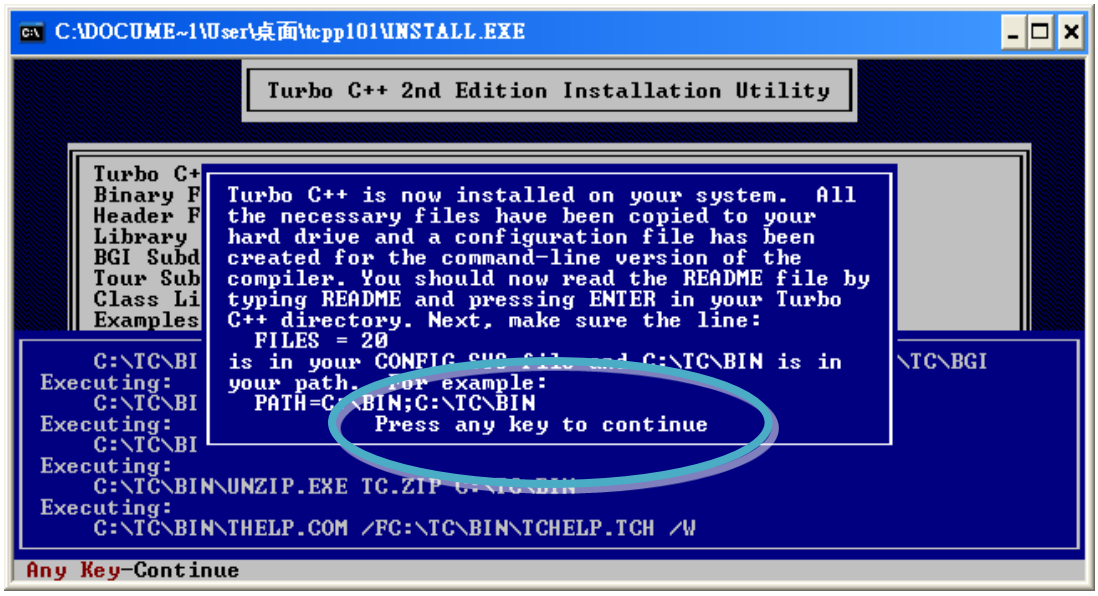


Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

Step 5: Select “Start Installation” to begin the install process

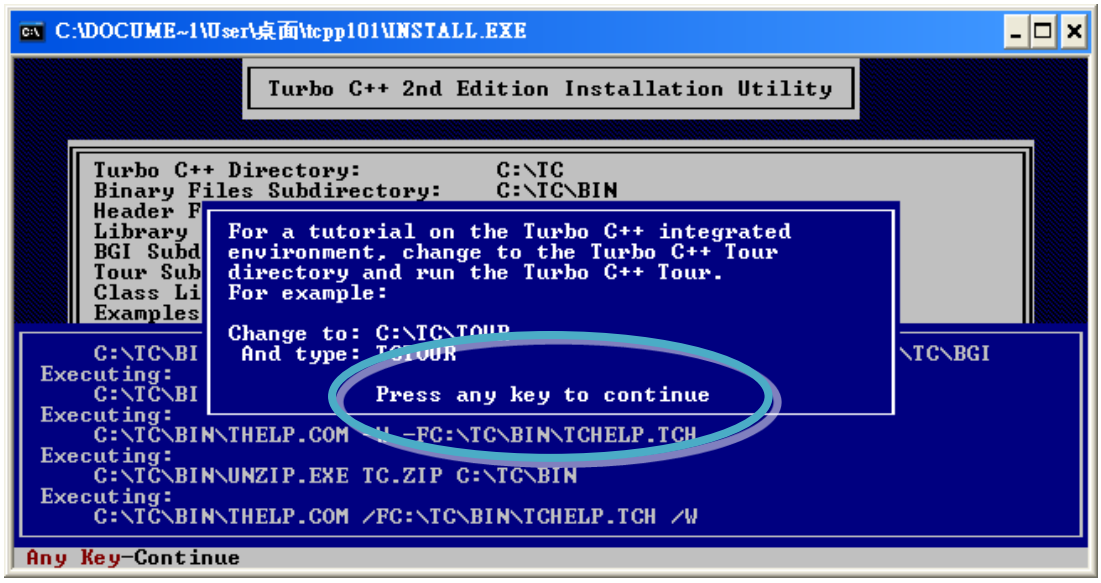

C:\DOCUME~1\User\桌面\tcpp101\INSTALL.EXE
Turbo C++ 2nd Edition Installation Utility
Turbo C++ Directory: C:\TC
Binary Files Subdirectory: C:\TC\BIN
Header Files Subdirectory: C:\TC\INCLUDE
Library Subdirectory: C:\TC\LIB
BGI Subdirectory: C:\TC\BGI
Tour Subdirectory: C:\TC\TOUR
Class Library Subdirectory: C:\TC\CLASSLIB
Examples Subdirectory: C:\TC\EXAMPLES
Install Tour: Yes
Yes
Memory Models... [S M C L H]
Start Installation
Description
Selecting this option will begin copying files to your hard drive into the directories specified above.
F1-Help F9-Start the installation ENTER-Select ESC-Previous

Step 6: Press any key to continue


C:\DOCUME~1\User\桌面\tcpp101\INSTALL.EXE
Turbo C++ 2nd Edition Installation Utility
Turbo C++ is now installed on your system. All the necessary files have been copied to your hard drive and a configuration file has been created for the command-line version of the compiler. You should now read the README file by typing README and pressing ENTER in your Turbo C++ directory. Next, make sure the line:
FILES = 20
is in your CONFIG.SYS file and C:\TC\BIN is in your path. For example:
PATH=C:\BIN;C:\TC\BIN
Press any key to continue
Executing:
C:\TC\BIN\UNZIP.EXE TC.ZIP
Executing:
C:\TC\BIN\THELP.COM /FC:\TC\BIN\TCHELP.TCH /W
Any Key-Continue

Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

Step 7: Press any key to continue


C:\DOCUME~1\User\桌面\tcpp101\INSTALL.EXE
Turbo C++ 2nd Edition Installation Utility
Turbo C++ Directory: C:\TC
Binary Files Subdirectory: C:\TC\BIN
For a tutorial on the Turbo C++ integrated environment, change to the Turbo C++ Tour directory and run the Turbo C++ Tour.
For example:
Change to: C:\TC\TOUR
Press any key to continue
Executing:
C:\TC\BIN\THELP.COM /FC:\TC\BIN\TCHELP.TCH
Executing:
C:\TC\BIN\UNZIP.EXE TC.ZIP C:\TC\BIN
Executing:
C:\TC\BIN\THELP.COM /FC:\TC\BIN\TCHELP.TCH /W
Any Key-Continue

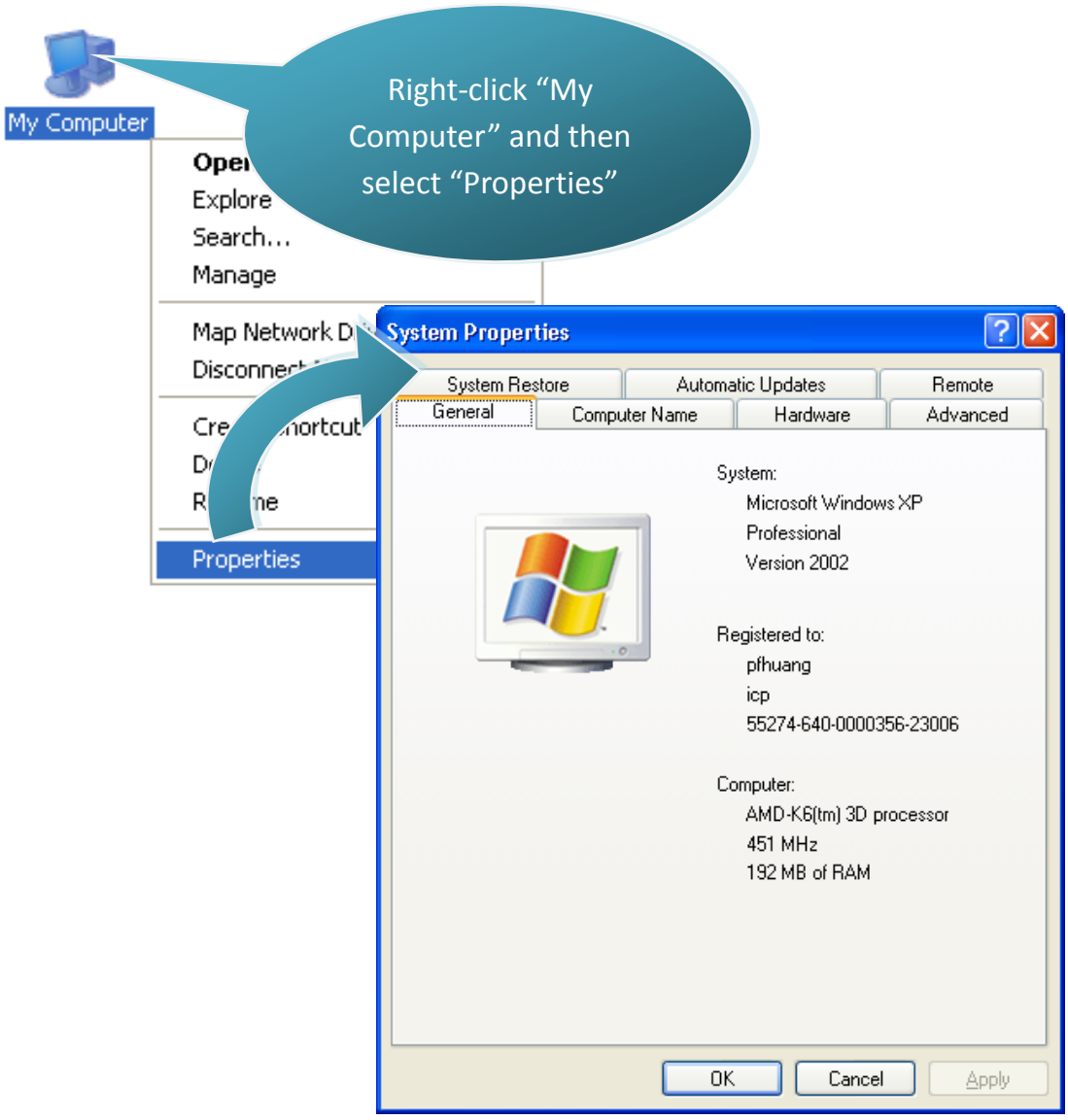
Step 8: Installation is complete

Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

3.1.2. Setting up the environment variables

After installing the compiler, several compilers will be available from the Windows Command line. You can set the path environment variable so that you can execute this compiler on the command line by entering simple names, rather than by using their full path names.

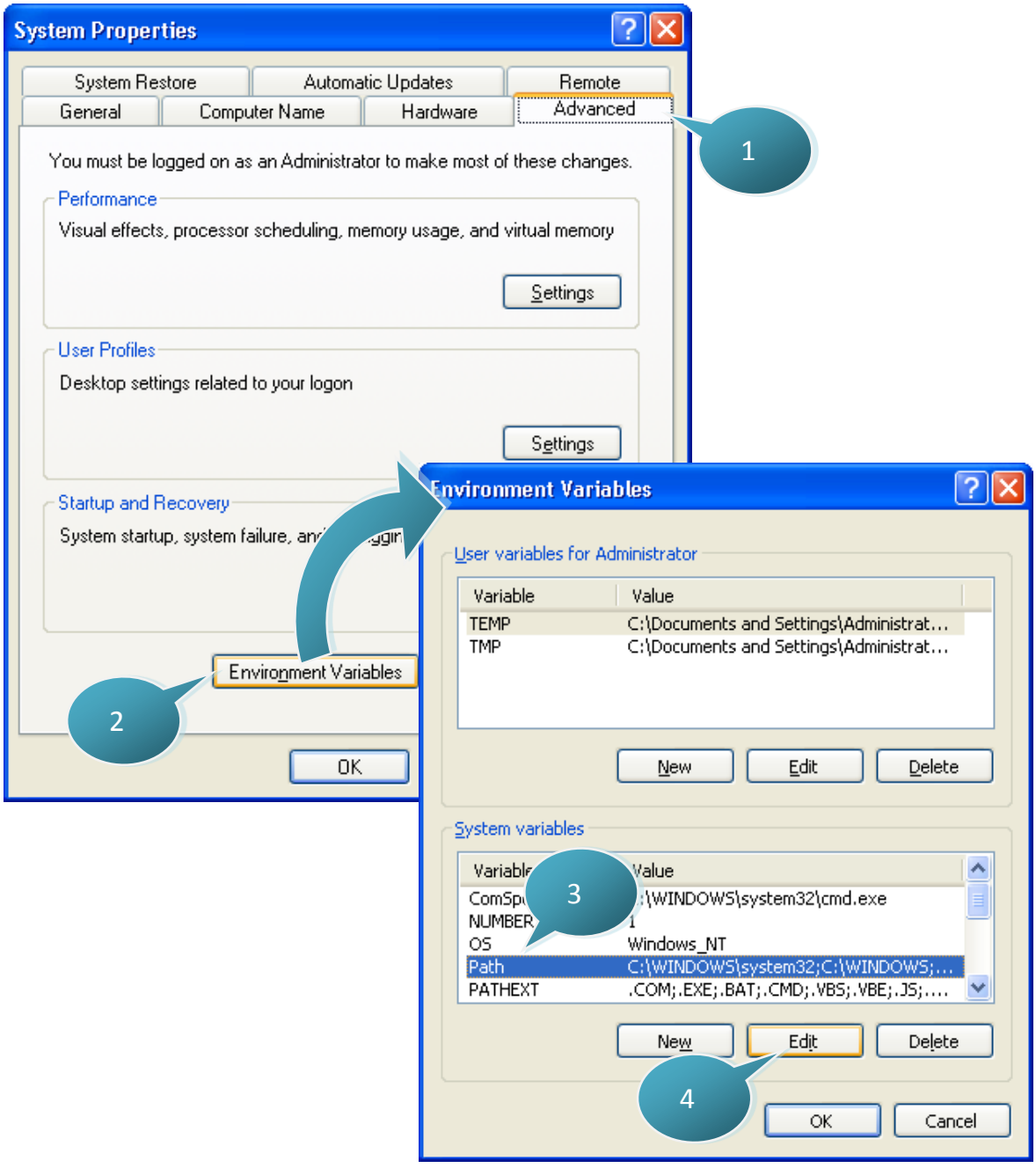
Step 1: Right click on the “My Computer” icon on your desktop and select the “Properties” menu option



Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

Step 2: On the “System Properties” dialog box, click the “Environment Variables” button located under the “Advanced” sheet

Step 3: On the “Environment Variables” dialog box, click the “Edit” button located in the “System variables” option



Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

Step 4: Add the target directory to the end of the variable value field

A semi-colon is used as the separator between variable values.

For example, ”;c:\TC\BIN\;c:\TC\INCLUDE\”

Step 5: Restart the computer to allow your changes to take effect

Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

3.2. µPAC-5000 APIs

There are several APIs for customizing the standard features and integrating with other applications, devices and services.

For more detailed information regarding µPAC-5000 APIs, please refer to
CD:\NAPDOS\upac-5000\Demo\basic\Lib\01_Lib_Update_History_20100507.txt
http://ftp.icpdas.com/pub/cd/8000cd/napdos/upac-5000/demo/basic/lib/01_Lib_Update_History_20100507.txt

Before creating the application, ensure them that you have installed. If they are not installed, please refer to “section 2.2. Software Installation”.

Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

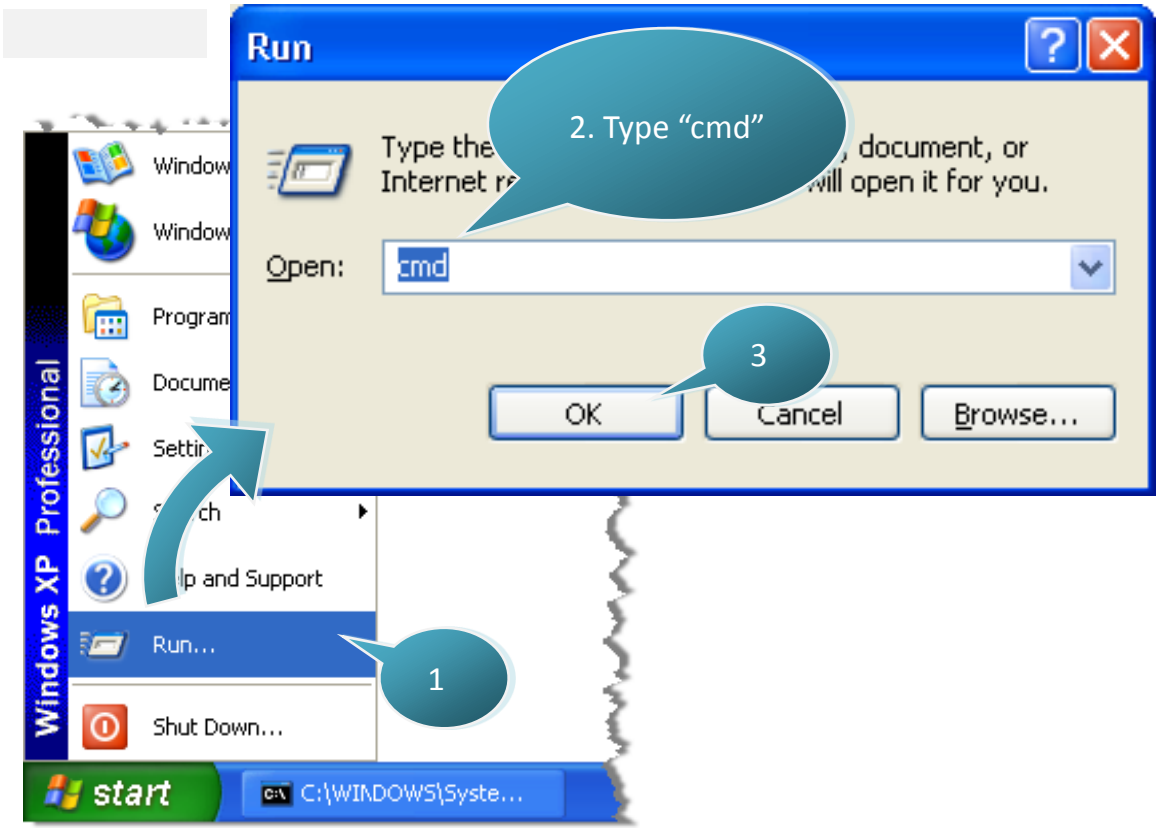
3.3. First Program in µPAC-5000

Here we assume you have installed the Turbo C++ 1.01 (as the section “3.1. C Compiler Installation”) and the µPAC-5000 APIs (as the section “2.2. Software Installation”) under the C driver root folder.

Below are step-by-step instructions for writing your first program.

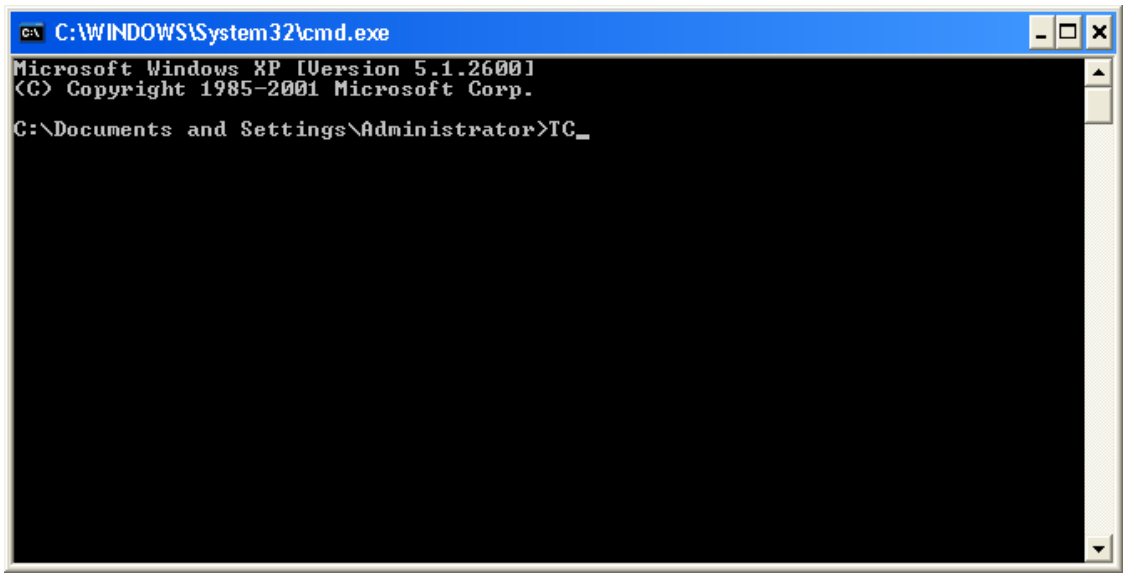
Step 1: Open a MS-DOS command prompt

i. Select “Run” from the “Start” menu
ii. On the “Run” dialog box, type “cmd”
iii. Click the “OK” button

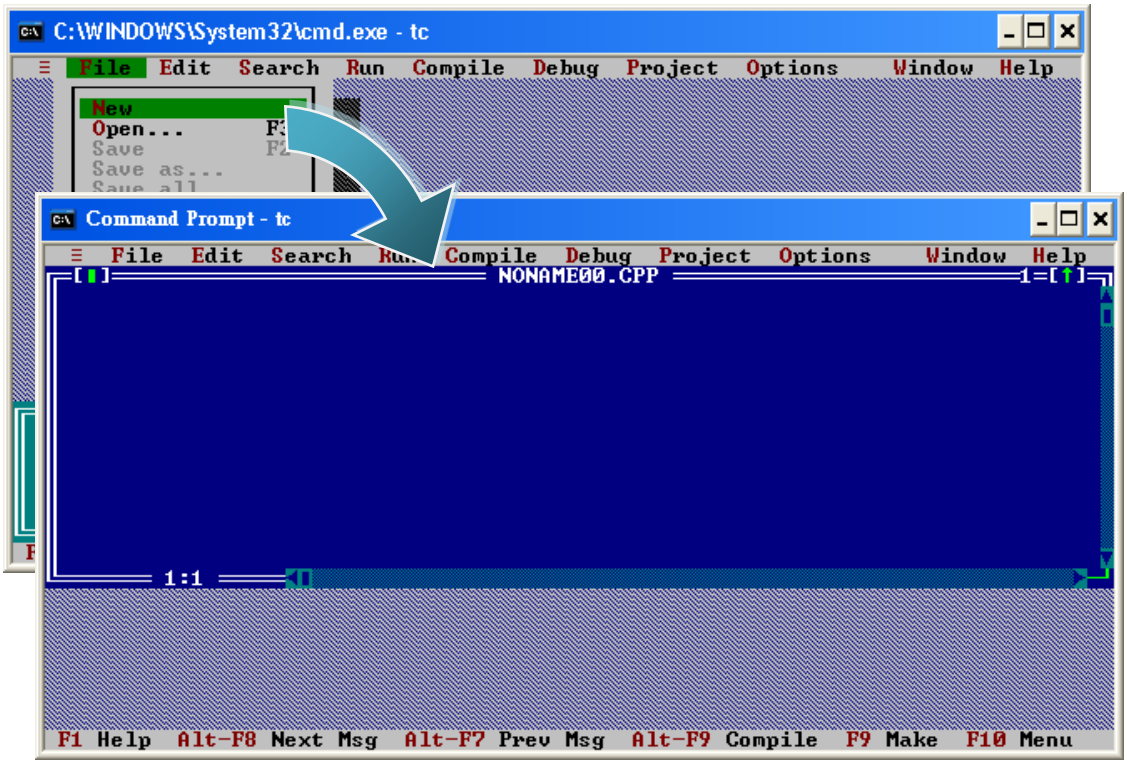


Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

Step 2: At the command prompt, type “TC” and then press “Enter”



Step 3: Select “New” from the “File” menu to create a new source file


C:\WINDOWS\System32\cmd.exe - tc
File Edit Search Run Compile Debug Project Options Window Help
New
Open... F3
Save F2
Save as...
Command Prompt - tc
File Edit Search Compile Debug Project Options Window Help
NONAME00.CPP
1:1
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt-F9 Compile F9 Make F10 Menu

Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

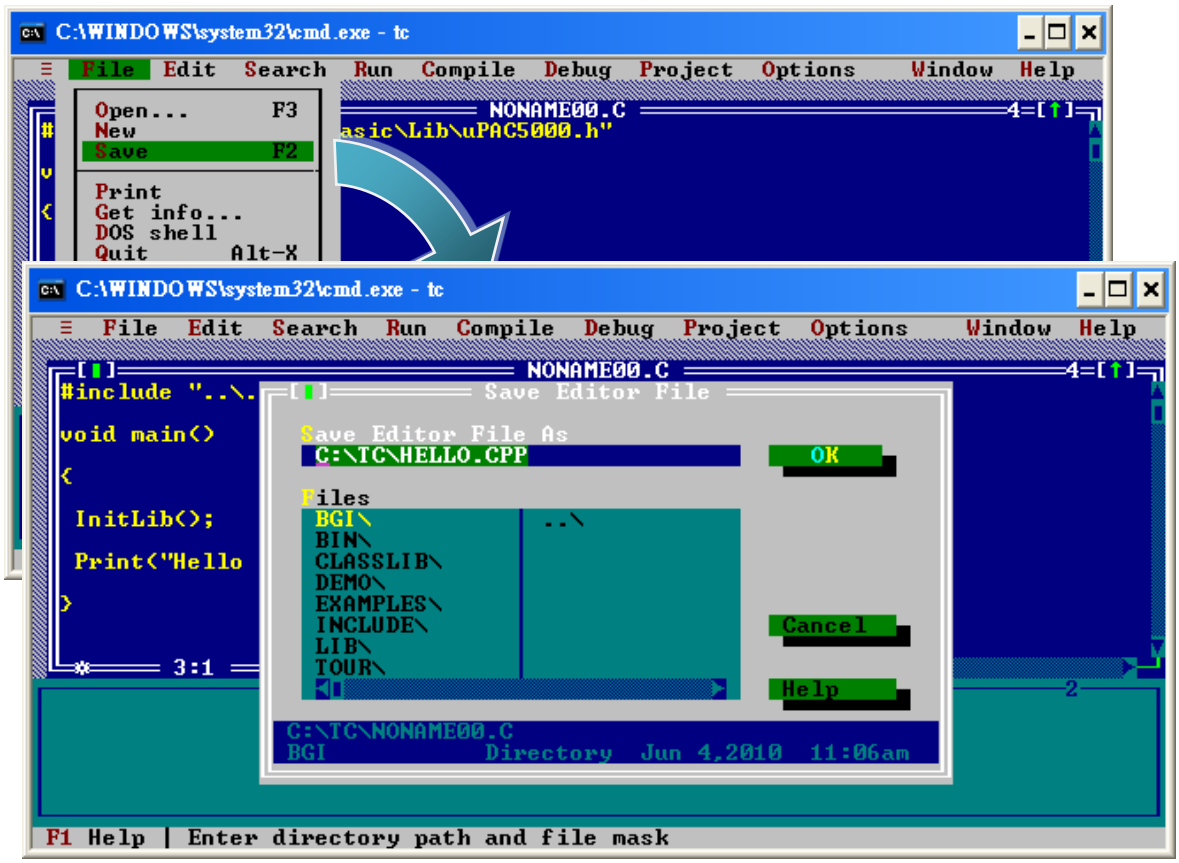
Step 4: Type the following code. Note that the code is case-sensitive

```
#include "..\..\Demo\basic\Lib\uPAC5000.h"
/* Include the header file that allows uPAC5000.lib functions to be used */

void main(void)
{
  InitLib();      /* Initiate the uPAC5000 library */
  Print("Hello world!\r\n");      /* Print the message on the screen */
}
```


Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

Step 5: Save the source file

i. Select “Save” from the “File” menu
ii. Type the file name “Hello”
iii. Select “OK”



Tips & Warnings

You can write the code as shown below with your familiar text editor or other tools; please note that

you must save the source code under a filename that terminates with the extension “C”.

Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

Step 6: Create a project (*.prj)

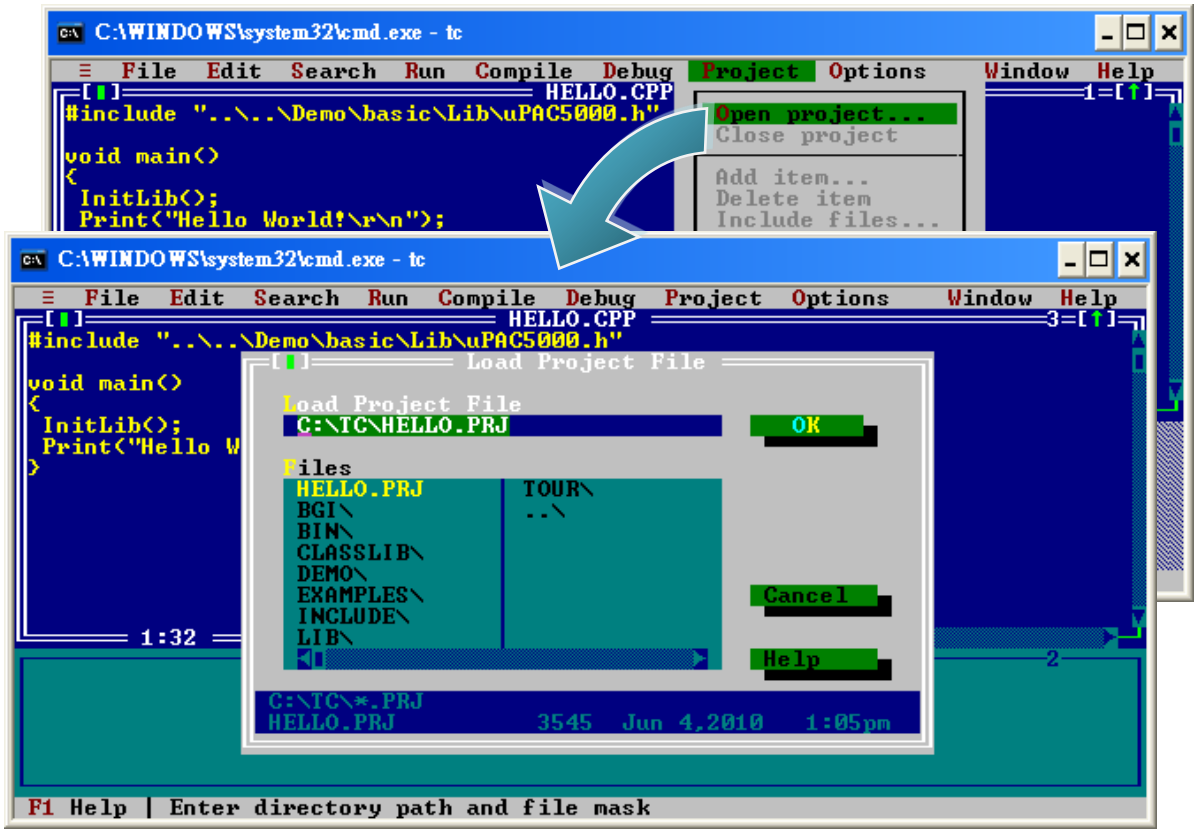
i. Select “Open project…” from the “Project” menu

ii. Type the project name “Hello”

iii. Select “OK”

iv. Select “Add”

v. Select “Done” to exit



Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

Step 8: Add the necessary function libraries to the project (*.lib)

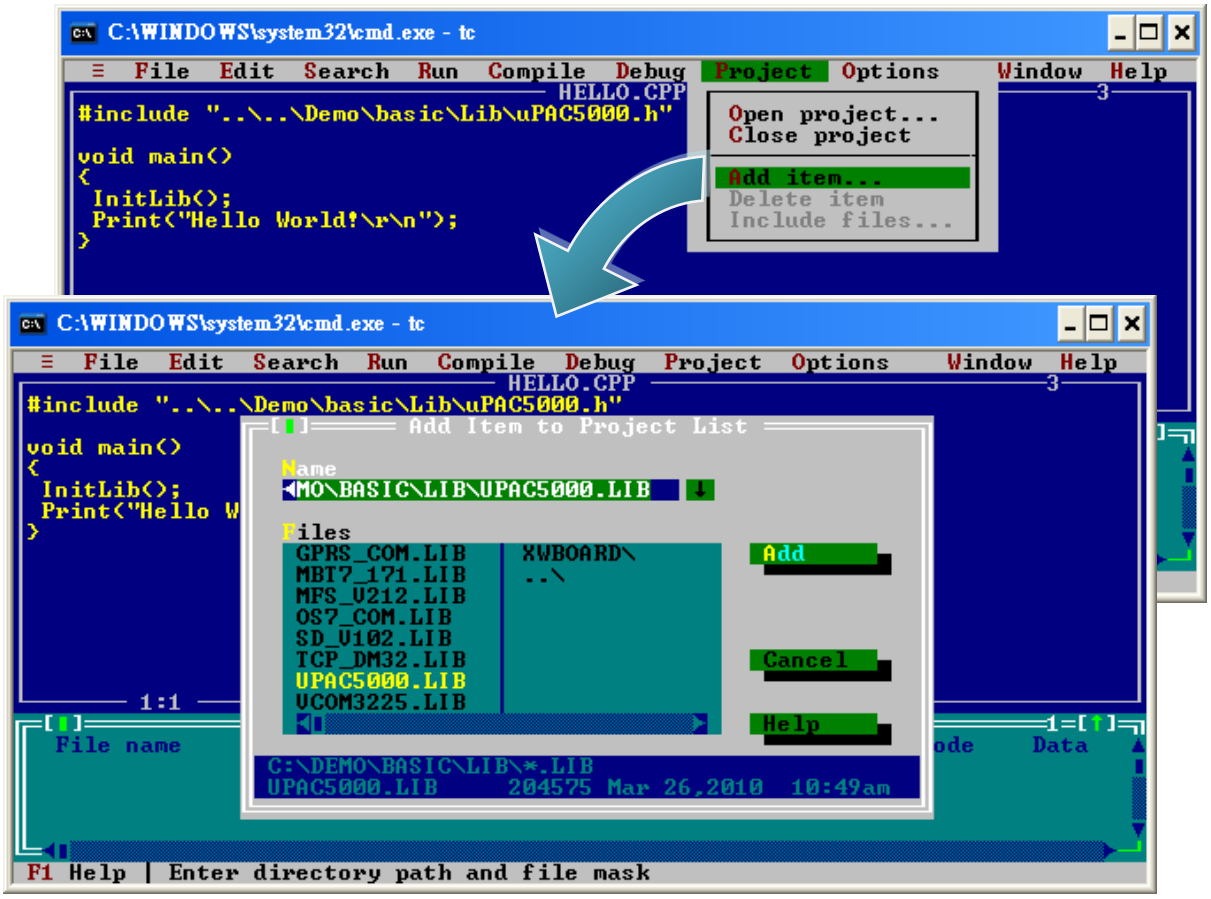
i. Select “Add item…” from the “Project” menu

ii. Type “ *.LIB ” to display a list of all available function libraries

iii. Choose the function libraries you require

iv. Select “Add”

v. Select “Done” to exit



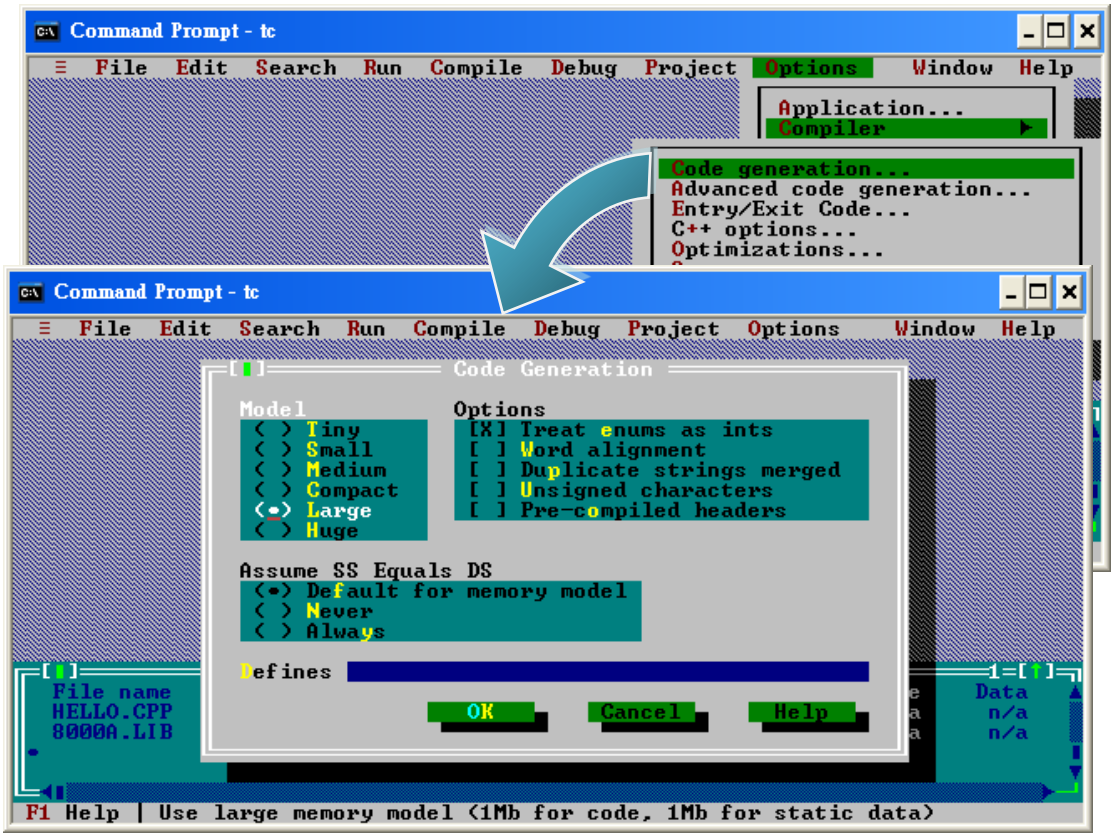
Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

Step 9: Set the memory model to large

i. Select “Compiler” from the “Options” menu and then select “Code generation…”

ii. On “Model” option, select “Large”

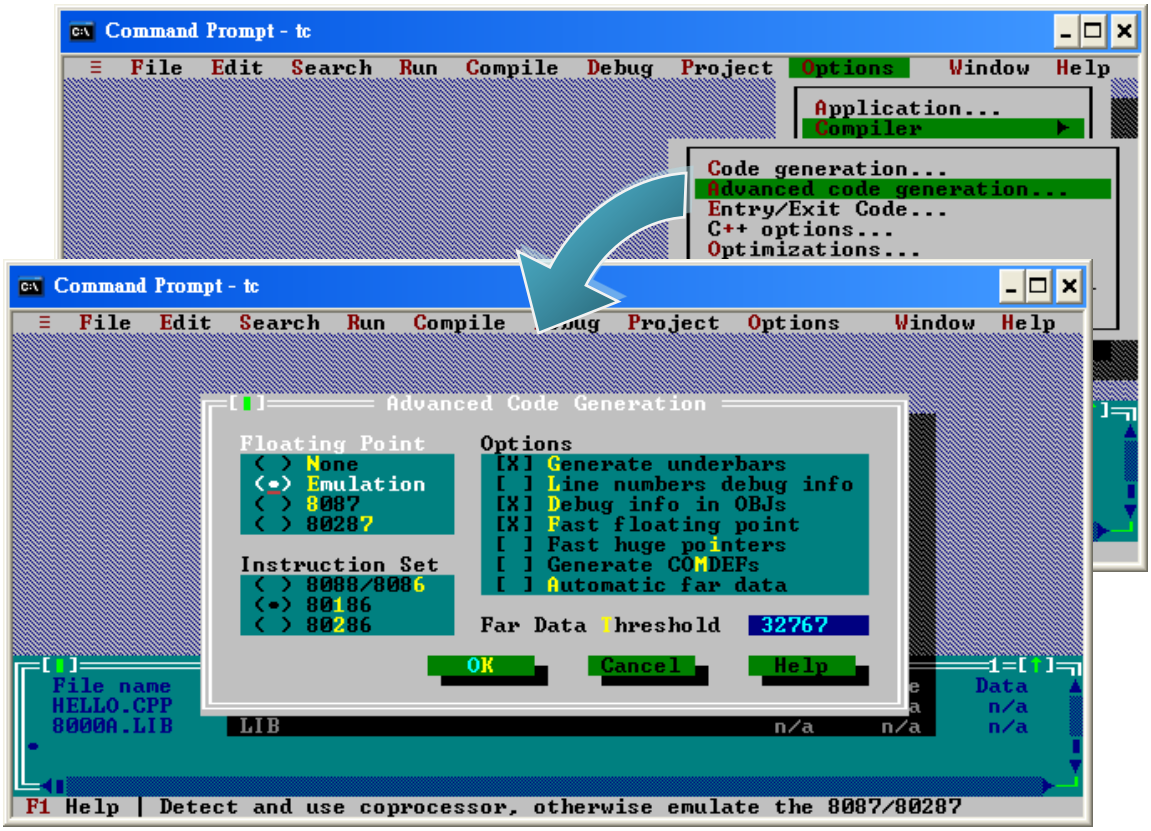
iii. Select “OK”



Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

Step 10: Set the memory model to large

i. Select “Compiler” from the “Options” menu and then select “Advanced code generation…”
ii. On “Floating Point” option, select “Emulation”
iii. On “Instruction Set” option, select “80186”
iv. Select “OK”



Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

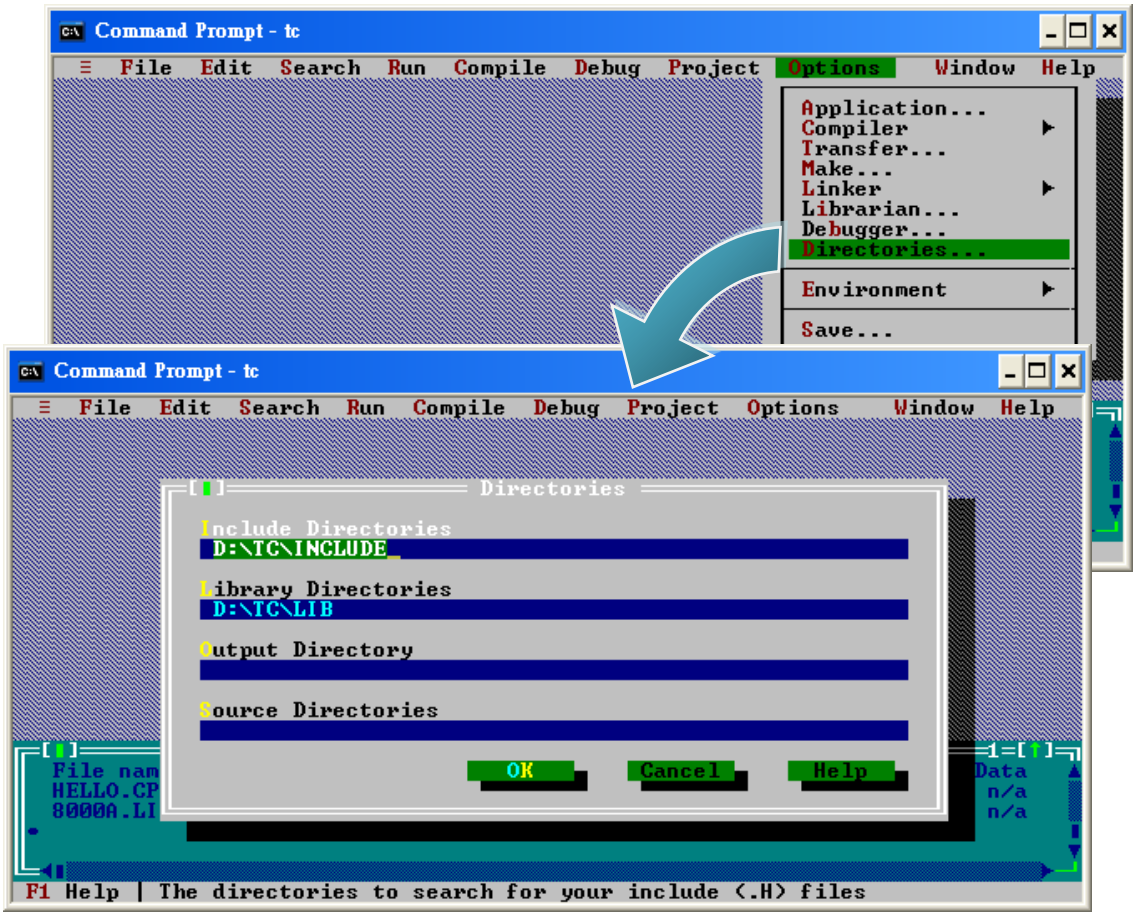
Step 11: Set the memory model to large

i. Select “Directories…” from the “Options” menu

ii. On “Include Directories” option, specify the header file

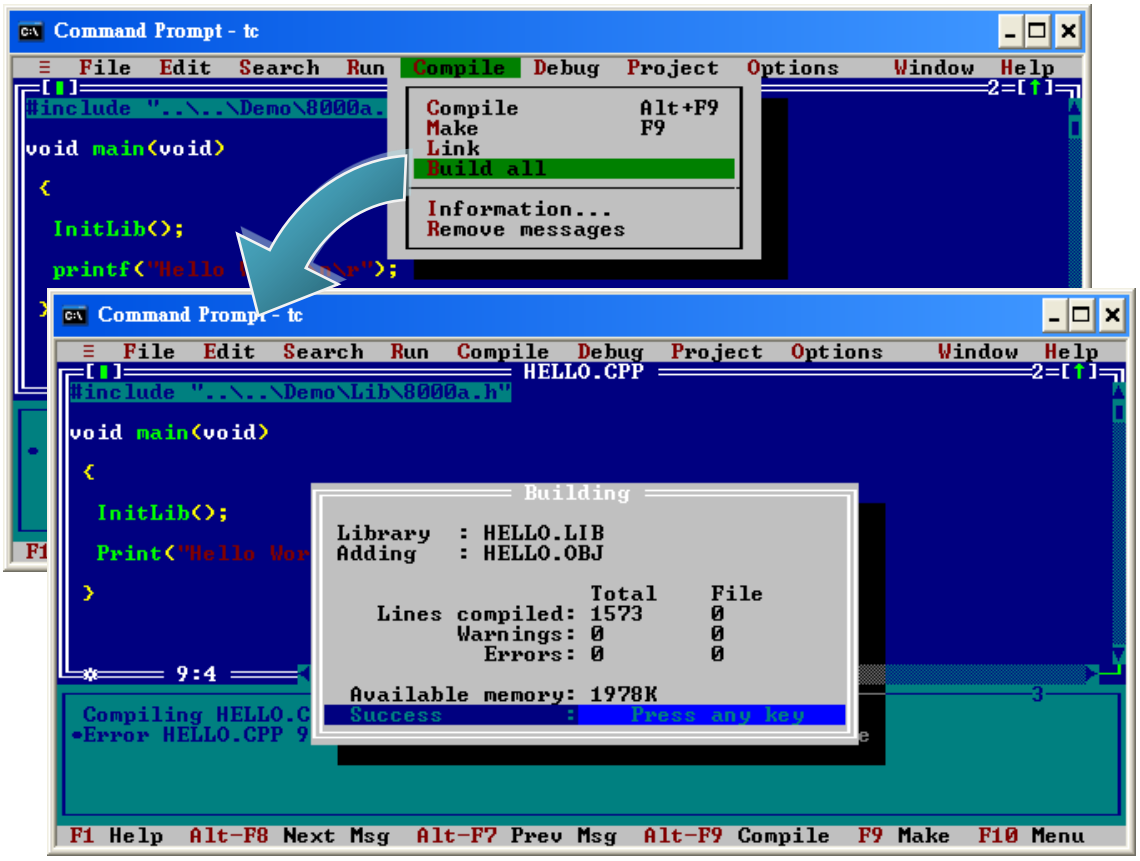
iii. On “Library Directories” option, specify the function library file

iv. Select “OK”



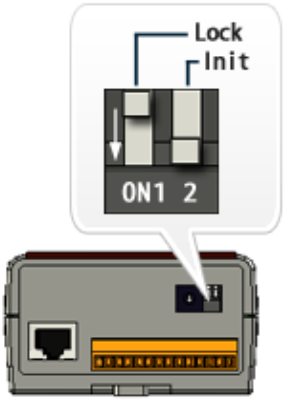
Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

Step 12: Select “Build all” from the “Compile” menu to build the project



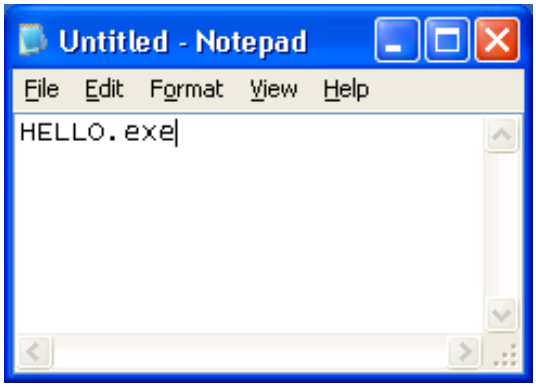
Step 13: Configure the operating mode

Make sure the switch of the Lock placed in the “OFF” position, and the switch of the Init placed in the “ON” position.



Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

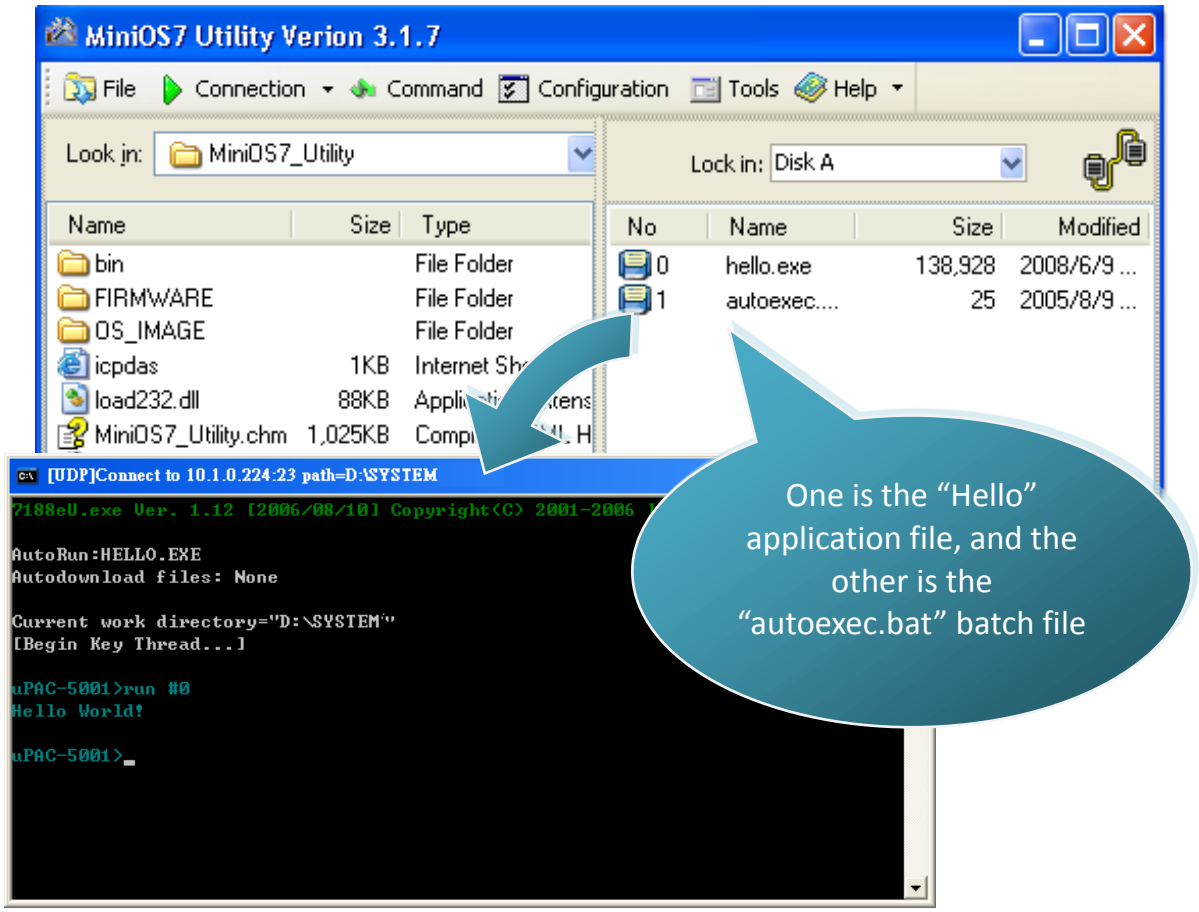
Step 14: Create an autoexec.bat file



i. Open the “Notepad”
ii. Type the “HELLO.exe”
iii. Save the file as autoexec.bat

Step 15: Upload programs to µPAC-5000 using MiniOS7 Utility

For more detailed information about this process, please refer to section “2.4.1. Establishing a connection”



Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

4. APIs and Demo References

There are several APIs and demo programs that have been designed for µPAC-5000.
You can examine the APIs and demo source code, which includes numerous functions and comments, to familiarize yourself with the MiniOS7 APIs and quickly develop your own applications quickly by modifying these demo programs.

The following table lists the APIs grouped by functional category.

API Description	Header File	Library
CPU driver	uPAC5000.h	uPAC5000.lib
new version of COM port driver	OS_COM.h	OS_COM.lib
Ethernet driver	TCPIP32.h	TCP_DM32.Lib
microSD driver	microSD.h	sd_V102.lib
64MB Flash Disk driver	MFS.h	MFS_V212.lib
GPRS driver	GPRS_COM.h	GPRS_COM.lib
Xserver	VxComm.h	Vcom3225.Lib
Modbus driver	MBTCP.h	MBT7_171.lib
Xboard driver	XW107.h	XW107.lib

For more detailed information regarding µPAC-5000 APIs, please refer to
CD:\NAPDOS\upac-5000\Demo\basic\Lib\01_Lib_Update_History_20100507.txt
http://ftp.icpdas.com/pub/cd/8000cd/napdos/upac-5000/demo/basic/lib/01_Lib_Update_History_20100507.txt

Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

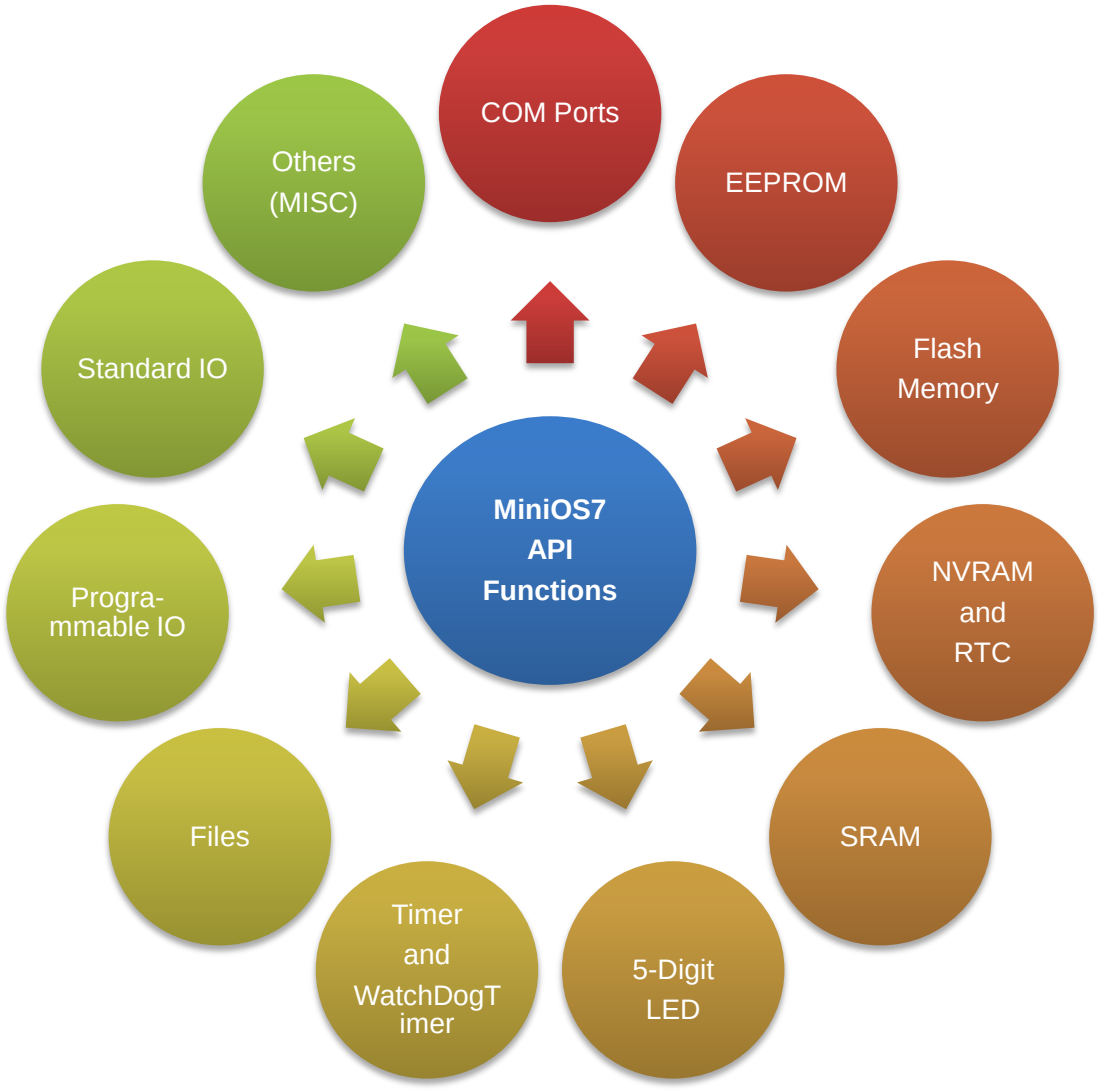
The following introduces the core API, MiniOS7 API, which is integrated into the uPAC5000 API set.

Functions Library – uPAC5000.lib

This file contains the MiniOS7 API (Application Programming Interface) and has hundreds of pre-defined functions related to µPAC-5000

Header File – uPAC5000.h

This file contains the forward declarations of subroutines, variables, and other identifiers used for the MiniOS7 API.

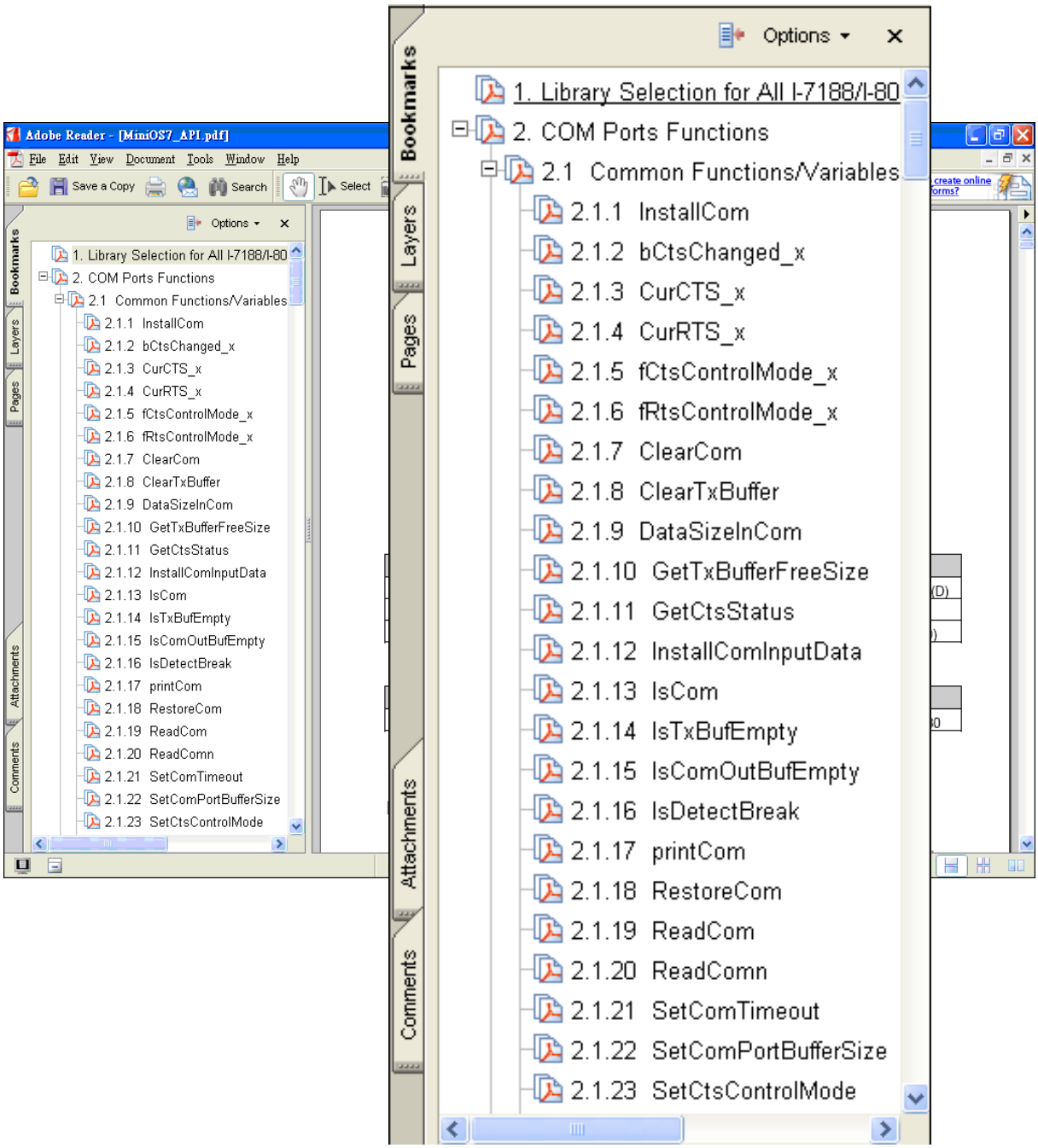


Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

For full usage information regarding the description, prototype and the arguments of the functions, please refer to the “MiniOS7 API Functions User Manual” located at:

CD:\Napdos\MiniOS7\Document

http://ftp.icpdas.com/pub/cd/8000cd/napdos/minios7/document/



Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

The following table lists the demo programs grouped by functional category.

Folder	Demo	Explanation
MISC	LED	Shows how to control the LED display.
	Rotary_Switch	Shows how to read the position of the switch.
64MB_Flash	Gets	Shows how to get a string from a file in the 64MB flash memory
	mFS_QA	Quality assurance program for the MiniOS7 File System. Including function test, read/write performance test.
	Puts	Shows how to write a string to a file in the 64MB flash memory.
	Utility	Utility for the MiniOS7 File System. Operations Include Dir, Read, Write, etc.
microSD	sd_qa	Shows how to connect and control the microSD
	sd_read	
	sd_util	
	sd_write	
modbus	MTDemo00	Shows how to use a standard Modbus driver to retrieve data.
	MTDemo01_Link_i7000	
	MTDemo03_Link_PLC	
	MTDemo05_Modbus_TCP_Master_1	
	MTDemo05_Modbus_TCP_Master_2	
XWboard	xw107	Shows how to connect and control the xw107 board.

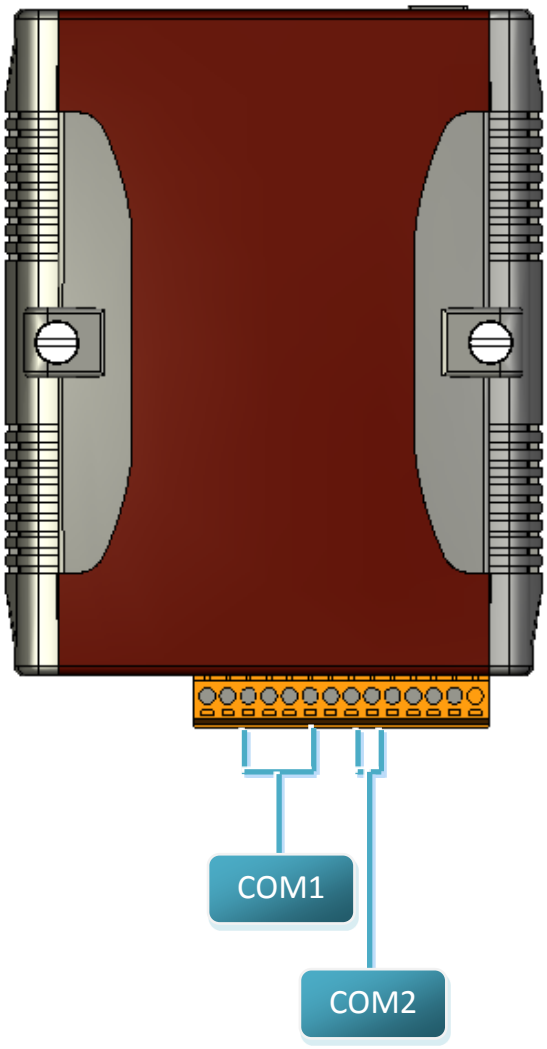
For more detailed information regarding µPAC-5000 APIs, please refer to CD:\NAPDOS\upac-5000\Demo\basic\
http://ftp.icpdas.com/pub/cd/8000cd/napdos/upac-5000/demo/basic/

Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

4.1. API for COM Port

The µPAC-5000 provides two built-in COM ports, COM1 and COM2.

- COM1 – A RS-232 port can use to connect to PC.
- COM2 – A RS-485 port in a point to point connection.



Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

4.1.1. Types of COM port functions

There are two types of functions below for using COM port.

1. MiniOS7 COM port functions
2. (C style) Standard COM port functions

Tips & Warnings

(C style) Standard COM port functions only can be used with the COM1, if you use the COM1 port, you'll have the alternative of MiniOS7 COM ports functions or (C style) Standard COM port functions. If you choose the ones, then another cannot be used.

Summarize the results of the comparison between MiniOS7 COM port functions and (C style) Standard COM port functions:

Types of Functions	COM Port	Buffer	Functions				
MiniOS7 COM port	1, 2, etc.	1 KB	1 KB	IsCom()	ToCom()	ReadCom()	printCom()
(C style) Standard COM port	1	512 Bytes	256 Bytes	Kbhit()	Puts() Putch()	Getch()	Print()

Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

4.1.2. API for MiniOS7 COM port

API for using COM ports

1. InstallCom()

Before any COM Port can be used, the driver must be installed by calling InstallCom().

2. RestoreCom()

If the program calls InstallCom(), the RestoreCom()must be called to restore the COM Port driver.

API for checking if there is any data in the COM port input buffer

3. IsCom()

Before reading data from COM port, the IsCom() must be called to check whether there is any data currently in the COM port input buffer.

API for reading data from COM ports

4. ReadCom()

After IsCom() confirms that the input buffer contains data, the ReadCom() must be called to read the data from the COM port input buffer.

API for sending data to COM ports

5. ToCom()

Before sending data to COM ports, the ToCom() must be called to send data to COM ports.

Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

For example, reading and receiving data through the COM1.

```
#include <stdio.h>
#include “upac5000.h”

void main(void)
{
 int quit=0, data;

 InitLib();      /* Initiate the upac5000 library */
 InstallCom(1, 115200, 8, 0, 1);      /* Install the COM1 driver */

 while(!quit)
 {
  if(IsCom(1))      /* Check if there is any data in the COM port input buffer */
  {
   data=ReadCom(1);      /* Read data from COM1 port */
   ToCom(1, data);      /* Send data via COM1 port */
   if(data==’q’) quit=1;      /* If ‘q’ is received, exit the program */
  }
 }
 RestoreCom(1);      /* Uninstall the COM1 driver */
}
```


Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

6. printCom()

Functions such as printfCom() in the C library allow data to be output from COM ports.

For example, showing data from the COM1 port.

```
#include <stdio.h>
#include “upac5000.h”

void main(void)
{
 int i;

 /* Initiate the upac5000 library */
 InitLib();
 InstallCom(1, 115200, 8, 0, 1);      /* Install the COM1 driver */
 for (i=0;i<10;i++)
 {
  printCom(1,”Test %d\n\r”, i);
 }
 Delay(10);      /* Wait for all data are transmitted to COM port */
 RestoreCom(1);
}
```


Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

4.1.3. API for standard COM port

The standard COM port is used to upload program from PC to the µPAC-5000.

Tips & Warnings

(C style) Standard COM port functions only can be used with the COM1 port, the following configurations of the COM1 port are fixed:

Baudrate = 115200 bps, Data format = 8 bits

Parity check = none, Start bit = 1, Stop bit = 1

API for checking if there is any data in the input buffer

1. Kbhit()

Before reading data from standard I/O port, the kbhit() must be called to check whether there is any data currently in the input buffer.

API for reading data from standard I/O port

2. Getch()

After kbhit() confirms that the input buffer contains data, the Getch() must be called to read data from the input buffer.

API for sending data to standard I/O port

3. Puts() – For sending a string

Before sending data to standard I/O port, the Puts() must be called to send data to COM Port..

Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

4. Putch( ) – For sending one character

Before sending data to standard I/O port, the Putch() must be called to send data to COM Port.

API for showing data from standard I/O port

5. Print()

Functions such as Print() in the C library allow data to be output from the COM port.

For example, reading and receiving data through COM1.

```
#include<stdio.h>
#include “upac5000.h”

void main(void)
{
 int quit=0, data;

 InitLib();      /* Initiate the upac5000 library */
 while(!quit)
 {
  if(Kbhit())      /* Check if any data is in the input buffer */
  {
   data=Getch();      /* Read data from COM1 */
   Putch(data);      /* Send data to COM1 */
   if(data==’q’) quit=1;      /* If ‘q’ is received, exit the program */
  }
 }
}
```


Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

For example, showing data through COM1.

```
#include <stdio.h>
#include “upac5000.h”

void main(void)
{
 int i;

 /* Initiate the upac5000 library */
 InitLib();
 for(i=0;i<10;i++)
 {
  Print(“Test %d\n\r”,i);
 }
}
```


Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

4.1.4. Port functions Comparison

For example, learning to show the ASCII code.

MiniOS7 COM port functions	Standard COM port functions

#include<stdio.h>
#include “upac5000.h”
 	
void main(void)
{
unsigned char item;
 	
InitLib();
InstallCom(1, 115200, 8, 0, 1);
printCom(1,”Hits any key.\n”);
printCom(1,”Hit the ESC to exit!\n”);
 	
for(;;)
{
if(IsCom(1))
{
item=ReadCom(1);
if(item==’q’)
{
return;
}
else
{
printCom(1,”----------\n\r”);
printCom(1,”char:”);
	#include<stdio.h>
	#include “upac5000.h”
	
	void main(void)
	{
	unsigned char item;
	
	InitLib();
	
	Print("Hits any key.\n");
	Print("Hits the ESC to exit !\n");
	
	for(;;)
	{
	if(kbhit())
	{
	item=Getch();
	if(item==’q’)
	{
	return;
	}
	else
	{
	Print(”----------\n\r”);
	Print(“char:“);

Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

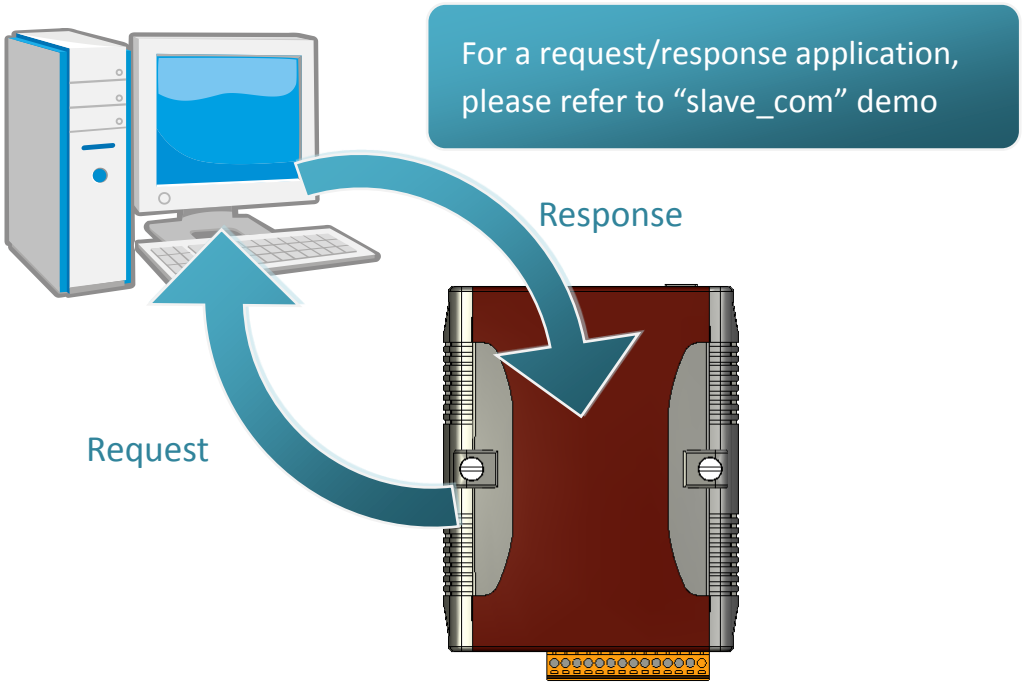
```
    ToCom(1,item);
    printCom(1,"\n\rASCII(%c)\n\r",item);
    printCom(1,"Hex(%02X)\n\r",item);
   }
  }
 }
 Delay(10);
 RestoreCom(1);
}
```

```
    Putch(item);
    Print("\n\rASCII(%c)\n\r",item);
    Print("Hex(%02X)\n\r",item);
   }
  }
 }


}
```


Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

4.1.5. Request/Response protocol define on COM port

Request/Response communication is very typical protocol architecture. If you want to design a command set of communication protocol as table below, you can refer to “slave_com” demo.

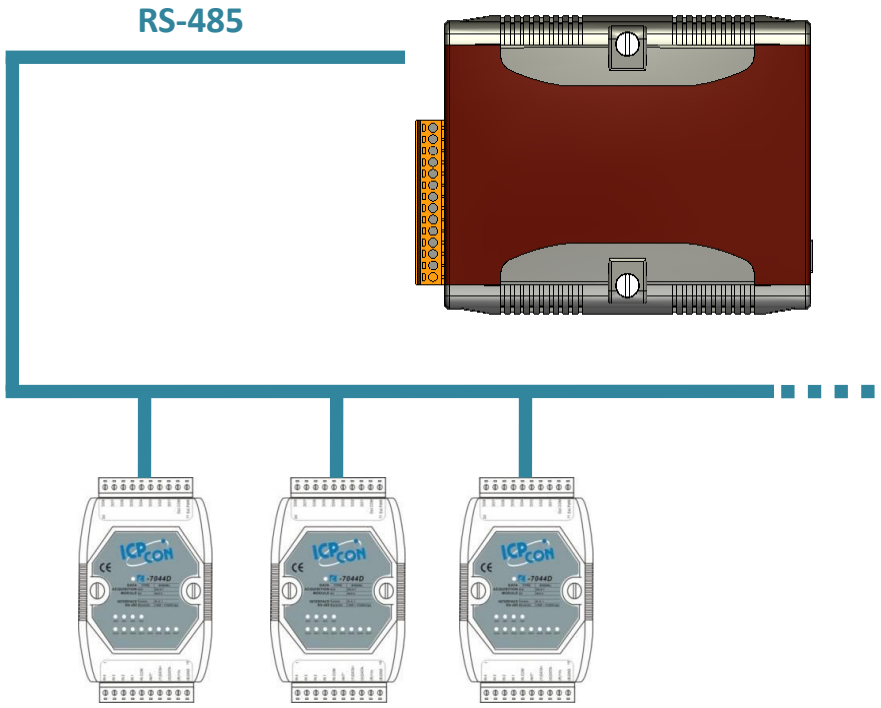


Request	Response
c1	Debug information: Command1 Command1
c2	Debug information: Command2 Command2
Q	Debug information: Quick program
Other command	Debug information: Unknown command

Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

4.2. API for I/O Modules

The µPAC-5000 equip a RS-485 communication interface, COM2, to access I-7K series I/O modules for a wide range of RS-485 network application, as shown below.



Steps to communicate with i-7K series I/O modules:

Step 1: Use Installcom() to install the COM port driver.
Step 2: Use SendCmdTo7000(2,…) to send commands
Step 3: Use ReceiveResponseFrom7000_ms() to get the response.
Step 4: Use RestoreCom() to restore the COM port driver

Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

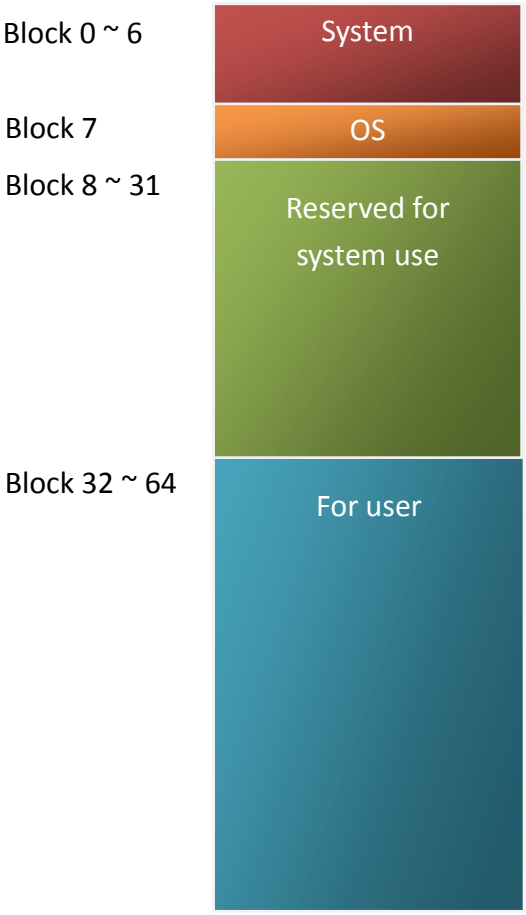
For example, to send a command '$01M' to I-7K I/O module for getting the module name.

```
#include <stdio.h>
#include "upac5000.h"

void main(void)
{
 unsigned char InBuf0[60];
 InitLib();      /* Initiate the upac5000 library */

 InstallCom(1,115200,8,0,1);      /* Install the COM1 driver */
 InstallCom(2,115200,8,0,1);      /* Install the COM2 driver */

 SendCmdTo7000(2,"$01M",0);      /* Send a command to COM2 */

 /* Timeout = 50ms, check sum disabled */
 ReceiveResponseFrom7000_ms(2,InBuf0,50,0);

 printCom(1,"Module Name = %s", InBuf0);
 Delay(10);      /* Wait for all data are transmitted to COM port */
 RestoreCom(1);      /* Uninstall the COM1 driver */

 RestoreCom(2);      /* Uninstall the COM2 driver */
}
```


Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

4.3. API for EEPROM

- The EEPROM contains 64 blocks (block 0 ~ 63), and each block has 256 bytes (address 0 ~ 255), with a total size of 16,384 bytes (16K) capacity.
- The default mode for EEPROM is write-protected mode.
- The system program and OS are stored in EEPROM that are allocated as shown below.



API for writing data to the EEPROM

1. EE_WriteEnable()

Before writing data to the EEPROM, the EE_WriteEnable() must be called to write-enable the EEPROM.

2. EE_WriteProtect()

After the data has finished being written to the EEPROM, the EE_WriteProtect() must be called to in order to write-protect the EEPROM.

3. EE_MultiWrite()

After using the EE_WriteEnable() to write-enable EEPROM, the EE_MultiWrite()must be called to write the data.

API for reading data from the EEPROM

4. EE_MultiRead()

The EE_WriteEnable() must be called to read data from the EEPROM no matter what the current mode is.

Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

For example, to write data to block1, address 10 of the EEPROM:

```
#include <stdio.h>
#include “upac5000.h”

void main(void)
{
 int data=0x55, data2;

 InitLib();      /* Initiate the upac5000 library */
 EE_WriteEnable();
 EE_MultiWrite(1,10,1,&data);
 EE_WriteProtect();

 EE_MultiRead(1,10,1,&data2);      /* Now data2=data=0x55 */
}
```


Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

4.4. API for Flash Memory

- The µPAC-5000 module contains 512 Kbytes of Flash memory.
- MiniOS7 uses the last 64K bytes; the other parts of the memory are used to store user programs or data.
- Each bit of the Flash memory only e can be written from 1 to 0 and cannot be written from 0 to 1.
- Before any data can be written to the Flash memory, the flash must be erased, first which returns all data to 0xFF, meaning that all data bits are set to “1”. Once there is completed, new data can be written.

Free: 448 K bytes
MiniOS7: 64 K bytes
Total Size: 512 K bytes

API for writing data to the Flash Memory

1. FlashWrite()

The FlashWrite() must be called to write data to the Flash Memory.

API for reading data from the Flash Memory

2. FlashRead()

The FlashRead() must be called to read data from the Flash Memory.

Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

For example, to write an integer to segnment 0xD000, offset 0x1234 of the Flash memory.

```
#include <stdio.h>
#include “upac5000.h”

void main(void)
{
int data=0xAA55, data2;
char *dataptr;
int *dataptr2;

InitLib();      /* Initiate the upac5000 library */
dataptr=(char *)&data;
FlashWrite(0xd000,0x1234, *dataptr++);
FlashWrite(0xd000,0x1235, *dataptr);

/* Read data from the Flash Memory (method 1) */
dataprt=(char *)&data2;
*dataptr=FlashRead(0xd000,0x1234);
*(dataptr+1)=FlashRead(0xd000,0x1235);

/* Read data from the Flash Memory (method 2) */
dataptr2=(int far *)_MK_FP(0xd000,0x1234);
data=*data;
}
```


Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

4.5. API for NVRAM

- The µPAC-5000 equip an RTC (Real Time Clock), 31 bytes of NVRAM can be used to store data.
- NVRAM is SRAM, but it uses battery to keep the data, so the data in NVRAM does not lost its information when the module is power off.
- NVRAM has no limit on the number of the re-write times. (Flash and EEPROM both have the limit on re-write times) If the leakage current is not happened, the battery can be used 10 years.

API for writing data to the NVRAM

1. WriteNVRAM()

The WriteNVRAM() must be called in order to write data to the NVRAM.

API for reading data from the NVRAM

2. ReadNVRAM()

The ReadNVRAM() must be called in order to write data to the NVRAM.

Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

For example, use the following code to write data to the NVRAM address 0.

```
#include <stdio.h>
#include “upac5000.h”

void main(void)
{
 int data=0x55, data2;
 InitLib();      /* Initiate the upac5000 library */
 WriteNVRAM(0,data);
 data2=ReadNVRAM(0);      /* Now data2=data=0x55 */
}
```

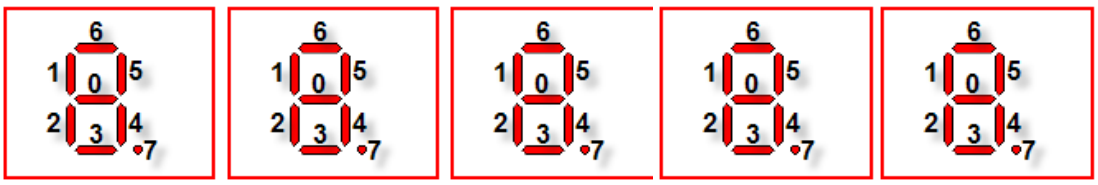
For example, the following can be used to write an integer (two bytes) to NVRAM.

```
#include <stdio.h>
#include “upac5000.h”

void main(void)
{
int data=0xAA55, data2;
char *dataptr=(char *)&data;

InitLib();      /* Initiate the upac5000 library */
WriteNVRAM(0, *dataptr);      /* Write the low byte */
WriteNVRAM(1, *dataptr+1);      /* Write the high byte */
dataptr=(char *) &data2;
*dataptr=ReadNVRAM(0);      /* Read the low byte */
(*dataptr+1)=ReadNVRAM(1);      /* Read the high byte */
}
```


Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

4.6. API for 5-Digital LED

The µPAC-5000 contains a 5-Digit 7-SEG LED with a decimal point on the left-hand side of each digit, which be used to display numbers, IP addresses, time, and so on.



API for starting the 5-Digit 7-SEG LED

1. Init5DigitLed()

Before using any LED functions, the Init5DigitLed() must be called to initialize the 5-Digit 7-SEG LED.

API for displaying a message on the 5-Digit 7-SEG LED

2. Show5DigitLed()

After the Init5DigitLed() is used to initialize the 5-Digit 7-SEG LED, the Show5DigitLed() must be called to display information on the 5-Digits 7-SEG LED.

Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

For example, use the following code to display “8000E” on the 5-Digit 7-SEG LED.

```
#include <stdio.h>
#include “upac5000.h”

void main(void)
{
 InitLib();      /* Initiate the upac5000 library */

 Init5DigitLed();
 Show5DigitLed(1,8);
 Show5DigitLed(2,0);
 Show5DigitLed(3,0);
 Show5DigitLed(4,0);
 Show5DigitLed(5,14);      /* The ASCII code for the letter ‘E’ is 14 */

}
```


Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

4.7. API for Timer

- The µPAC-5000 can support a single main time tick, 8 stop watch timers and 8 counts down timers.
- The µPAC-5000 uses a single 16-bit timer to perform these timer functions, with a timer accuracy of 1 ms..

API for starting the Timer

1. TimerOpen()

   Before using the Timer functions, the TimerOpen() must be called at the beginning of the program.

API for reading the Timer

2. TimerResetValue()

   Before reading the Timer, the TimerResetValue() must be called to reset the main time ticks to 0.

3. TimerReadValue()

   After the TimerResetValue() has reset the main time ticks to 0, the TimerReadValue() must be called to read the main time tick.

API for stopping the Timer

4. TimerClose()

   Before ending the program, the TimerClose() must be called to stop the Timer.

Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

For example, the following code can be used to read the main time ticks from 0

```
#include <stdio.h>
#include “upac5000.h”

void main(void)
{
 Unsigned long time iTime;

 InitLib();      /* Initiate the upac5000 library */
 TimerOpen();
 While(!quit)
 {
  If(Kbhit())
  TimerResetValue();      /* Reset the main time ticks to 0 */

  iTime=TimerReadValue();      /* Read the main time ticks from 0 */
 }
 TimerClose();      /* Stop using the uPAC5000 timer function */
}
```


Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

4.8. API for WatchDog Timer (WDT)

- The µPAC-5000 equips the MiniOS7, the small-cored operating system. MiniOS7 uses the Timer 2 (A CPU internal timer) as system Timer. It is 16-bits Timer, and generate interrupt every 1 ms. So the accuracy of system is 1 ms.
- The Watch Dog Timer is always enabled, and the system Timer ISR (Interrupt Service Routine) refreshes it.
- The system is reset by WatchDog. The timeout period of WatchDog is 0.8 seconds.

API for refreshing WDT

1. EnableWDT()

   The WDT is always enabled, before user's programming to refresh it, the EnableWDT() must be called to stop refreshing WDT.

2. RefreshWDT()

   After EnableWDT() stop refreshing WDT, the RefreshWDT() must be called to refresh the WDT.

3. DisableWDT()

   After user's programming to refresh WDT, the DisableWDT() should be called to automatically refresh the WDT.

Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

For example, to refresh the Watchdog Timer.

```
#include <stdio.h>
#include “upac5000.h”

void main(void)
{
 Unsigned long time iTime;

 InitLib();      /* Initiate the upac5000 library */
 Enable WDT();
 While(!quit)
 {
  RefreshWDT();
  User_function();
 }
 DisableWDT();
}
```


Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

4.9. API for MFS (For µPAC-5000-FD series only)

Required library and header files:
MFS_V212.LIB and MFS.h

The µPAC-5000-FD series products equip an extra 64MB flash memory, the MFS is designed to read/write file from/to the 64MB flash memory.

For full usage information regarding the hardware supported, applications, and the specification, please refer to section “Appendix C. What is MiniOS7 File System (MFS)”

- Summarize of the MFS functions:

Function	Description
mfs_Init	Initialize the file system.
mfs_Stop	Allocated buffers are freed upon closing.
mfs_ResetFlash	Initialize the file system. All files will lose.
mfs_X600Fs_GetLibVersion	Gets the version number of function library.
mfs_GetLibDate	Gets the create date of function library.
mfs_GetFileNo	Gets the total number of files stored in the NAND Flash.
mfs_GetFreeSize	Gets the size of available space that can be used to append file.
mfs_GetBadSize	Gets the size of non-available space.
mfs_GetUsedSize	Gets the size of used space.
mfs_GetFileSize	Gets the size of file stored in the NAND Flash.
mfs_GetFileInfoByName	Uses the specified filename to retrieve file information.
mfs_GetFileInfoByNo	Uses the file number index to retrieve file information.
mfs_DeleteAllFiles	Delete all files stored in the NAND Flash.
mfs_DeleteFile	Delete one selected file that has been written to the NAND Flash.
mfs_OpenFile	1. Opens a file with a file name. 2. Creates a new file.
mfs_CloseFile	Closes a file with a file handle. All buffers associated with the stream are flushed before closing.

Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

Function	Description
mfs_ReadFile	Reads specified bytes of data from a file.
mfs_WriteFile	Appends specified bytes of data to a file.
mfs_Getc	Gets a character from a file.
mfs_Putc	Outputs a character data to the file.
mfs_Gets	Gets a string from a file.
mfs_Puts	Outs a string a file.
mfs_EOF	Macro that tests if end-of-file has been reached on a file.
mfs_Seek	Repositions the file pointer of a file.
mfs_Tell	Returns the current file pointer.
mfs_EnableWriteVerify	Enable the data verification. By default, the data verification is enabling.
mfs_DisableWriteVerify	Disable the data verification.

Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

API for starting 64MB flash memory

1. mfs_ Init()

Before using any MFS functions, the mfs_Init() must be called to initialize the 64MB flash memory.

2. mfs_Stop()

If the program calls the mfs_Init() to initialize the 64MB flash memory, the mfs_Stop() must be called to allocate buffers to free upon closing.

API for writing/reading files from the 64MB flash memory

3. mfs_OpenFile()

Before writing/reading data to/from the 64MB flash memory, the OpenFile() must be called to open the file.

4. mfs_CloseFile()

After the data has finished being written/read to/from the 64MB flash memory, the mfs_CloseFile() must be called to close the file with a file handle.

API for writing data to the 64MB flash memory

5. mfs_Puts()

After using the mfs_OpenFile() to open the file, the FlashRead() must be called to read data from the Flash Memory.

Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

For example, writing data to the 64MB flash memory:

```
#include <stdio.h>
#include “upac5000.h”
#include “MFS.h”

#define_DISK_A      0
#define_DISK_B      1

int    main(void)
{
 int iFileHandle, iRet;

 InitLib();      /* Initiate the upac5000 library */
 iRet=mfs_Init();
 if(iRet<=0) return;

 iFileHandle=mfs_OpenFile(_DISK_A,”Test.txt”,”w”);
 if(iFileHandle>0)
 {
  Print(“Write string to Test.txt…”);
  mfs_Puts(iFileHandle,”test mfs on 64MB flash”);
  mfs_CloseFile(iFileHandle);
  Print(“done”);
 }
 else
 Print(“Open file error\n\r”);
 mfs_Stop();
 return;
}
```


Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

6. mfs_Gets()

After using the mfs_OpenFile() to open the file, the mfs_Gets() must be called to read data from the 64MB flash memory.

For example, reading data from the 64MB flash memory:

```
#include <stdio.h>
#include “upac5000.h”
#include “MFS.h”

#define_DISK_A      0
#define_DISK_B      1

int    main(void)
{
 int iFileHandle, iRet;

 InitLib();      /* Initiate the upac5000 library */
 iRet=mfs_Init();
 if(iRet<=0) return;

 iFileHandle=mfs_OpenFile(_DISK_A,”Test.txt”,”r”);
 if(iFileHandle>0)
 {
  Print(“Read from Test.txt…\n\r”);
  iRet=mfs_Gets(iFileHandle,Data, 80);      /*max length is 80 bytes.*/
  if(iRet>0) Print(“Data=%s\n\r”,Data);

  mfs_CloseFile(iFileHandle);
```


Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

```
 Print("done");
}
else
Print("Open file error\n\r");
mfs_Stop();
return;
}
```

For more demo program about the Flash memory, please refer to:

CD:\NAPDOS\uPAC-5000\Demo\Basic\64MB_Flash\
http://ftp.Icpdas.com/pub/cd/8000cd/napdos/upac-5000/demo/basic/64mb_flash/

Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

4.10. API for microSD

Required library and header files:
SD_V102.LIB and microSD.h

The µPAC-5000 series can support one microSD card and the size can be 1GB or 2 GB.

- Summarize of the microSD functions:

Function	Description
pc_init	Initializes the microSD socket library
pc_open	1. Open an existing file and return a file handle 2. Creates a new file.
pc_close	Closes a file and release a file handle.
pc_read	Reads the specified file
pc_write	Writes the specified file
pc_seek	Moves the file pointer to relative offset from the current offset
pc_tell	Gets current offset of the file pointer
pc_eof	Checks whether the end-of-file is reached
pc_format	Formats the microSD card as FAT (FAT16)
pc_mkdir	Creates a directory or subdirectory
pc_rmdir	Removes an existing directory
pc_move	Renames an existing file or a directory, including the subdirectory
pc_del	Deletes the specified file
pc_deltree	Deletes the specified directory or subdirectory
pc_isdir	Checks whether the file is a directory
pc_isvol	Checks if is a volume
pc_size	Gets the size of the specified file
pc_set_cwd	Sets the current working directory
pc_get_cwd	Gets the pathname of the current working directory
pc_gfirst	Moves the pointer to the first element
pc_gnext	Moves the pointer to the next element
pc_gdone	Moves the pointer to the last element

Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

Function	Description
pc_get_freeSize_KB	Gets the free space of the SD memory card
pc_get_usedSize_KB	Gets the used space of the SD memory card
pc_get_totalSize_KB	Gets the total size of the SD memory card
pc_get_attributes	Gets the file attributes
pc_set_attributes	Sets the file attributes
pc_get_errno	Gets the error number

API for starting microSD

1. pc_ Init()

Before using any miscroSD functions, PC_Init() must be called to initialize the microSD.

API for enabling/disabling microSD

3. pc_open()

Before writing/reading data to/from the microSD card, PC_open() must be called to open the file.

4. pc_close()

After the data has finished being written/read to/from the microSD, PC_close() must be called to close the file with a file handle.

API for writing data to the microSD

5. pc_write()

After using PC_open() to open the file, PC_write() must be called to read data from the microSD.

Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

For example, writing data to the microSD

```
#include <string.h>
#include <stdio.h>
#include “upac5000.h”
#include “microSD.h”
{
  int fd, iRet;

  InitLib();
  If(pc_init())
  {
    Print(“Init microSD ok\n\r”);
  }
  else
  {
    Print(“Init microSD failed\n\r”);
    iRet=pc_get_errno();
    switch(iRet)
    {
      case PCERR_BAD_FORMAT:      //1
          Print("Error 01: format is not FAT\n\r");
          break;
      case PCERR_NO_CARD:      //2
          Print("Error 02: no microSD card\n\r");
          break;
      default:
          Print("Error %02d: unknow error\n\r",iRet);
    }
  }
```


Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

```
fd=pc_open("test.txt",(word)(PO_WRONLY|PO_CREAT|PO_APPEND),(word)(PS_IW
RITE|PS_IREAD));
  if(fd>=0)
  {
    pc_write(fd,"1234567890",10);      //write 10 bytes
    pc_close(fd);
  }
}
```

API for reading data from the microSD

6. pc_read()

After using PC_open() to open the file, PC_read() must be called to read data from the microSD.

For example, reading data from the microSD:

```
#include <string.h>
#include <stdio.h>
#include "upac5000.h"
#include "microSD.h"
{
  int fd, iRet;
  unsigned char Buffer[80];

  InitLib();
  If(pc_init())
  {
    Print("Init microSD ok\n\r");
  }
```


Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

```
  else
  {
    Print(“Init microSD failed\n\r”);
    iRet=pc_get_errno();
    switch(iRet)
    {
      case PCERR_BAD_FORMAT:      //1
          Print("Error 01: format is not FAT\n\r");
          break;
      case PCERR_NO_CARD:      //2
          Print("Error 02: no microSD card\n\r");
          break;
      default:
          Print("Error %02d: unknow error\n\r",iRet);
    }
  }
  fd=pc_open(“test.txt”,(word)(PO_RDONLY),(word)(PS_IWRITE|PS_IREAD));
  if(fd>=0)
  {
    iRet=pc_read(fd,Buffer,10);      //reads 10 bytes
    Buffer[10]=0;      //adds zero end to the end of the string.
    pc_close(fd);
    Print(“%s”,Buffer);
  }
}
```

For more demo program about the microSD, please refer to:

CD:\NAPDOS\uPAC-5000\Demo\Basic\microSD\

http://ftp.Icpdas.com/pub/cd/8000cd/napdos/upac-5000/demo/basic/microsd/

Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

Appendix A. What is MiniOS7?

MiniOS7 is an embedded ROM-DOS operating system design by ICP DAS. It is functionally equivalent to other brands of DOS, and can run programs that are executable under a standard DOS.

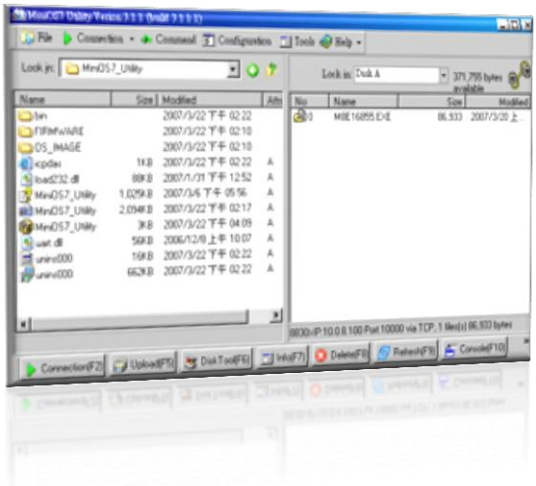
DOS (whether PC-DOS, MS-DOS or ROMDOS) is a set of commands or code that tells the computer how to process information. DOS runs programs, manages files, controls information processing, directs input and output, and performs many other related functions.

The following table compares the features between MiniOS7 and ROM-DOS:

Feature	MiniOS7	ROM-DOS
Power-up time	0.1 sec	4 ~ 5 sec
More compact size	< 64 K bytes	64 K bytes
Support for I/O expansion bus	Yes	No
Support for ASIC key	Yes	No
Flash ROM management	Yes	No
OS update (Upload)	Yes	No
Built-in hardware diagnostic functions	Yes	No
Direct control of 7000 series modules	Yes	No
Customer ODM functions	Yes	No
Free of charge	Yes	No

Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

Appendix B. What is MiniOS7 Utility?



MiniOS7 Utility is a tool for configuring, uploading files to all products embedded with ICP DAS MiniOS7.

Since version 3.1.1, the Utility can allow users remotely access the controllers (7188E, 8000E..., etc) through the Ethernet.

Functions

- Supported connection ways
  1. COM port connection (RS-232)
  2. Ethernet connection (TCP & UDP) (Supported since version 3.1.1)
- Maintenance
  1. Upload file(s)
  2. Delete file(s)
  3. Update MiniOS7 image
- Configuration
  1. Date and Time
  2. IP address
  3. COM port
  4. Disk size (Disk A, Disk B)
- Check product information
  1. CPU type
  2. Flash Size
  3. SRAM Size
  4. COM port number

  ..., etc.

Including frequently used tools

a. 7188XW
b. 7188EU
c. 7188E
---
d. SendTCP
e. Send232
f. VxComm Utility

Upload location:

http://ftp.icpdas.com/pub/cd/8000cd/napdos/minios7/utility/minios7_utility/

Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

Appendix C. What is MiniOS7 File System (MFS)?

MiniOS7 file system, MFS, offers a rugged alternative to mechanical storage systems. Designed for the 64MB NAND flash memory,

MFS implements a reliable file system with C language API for embedded data logger applications on MiniOS7.

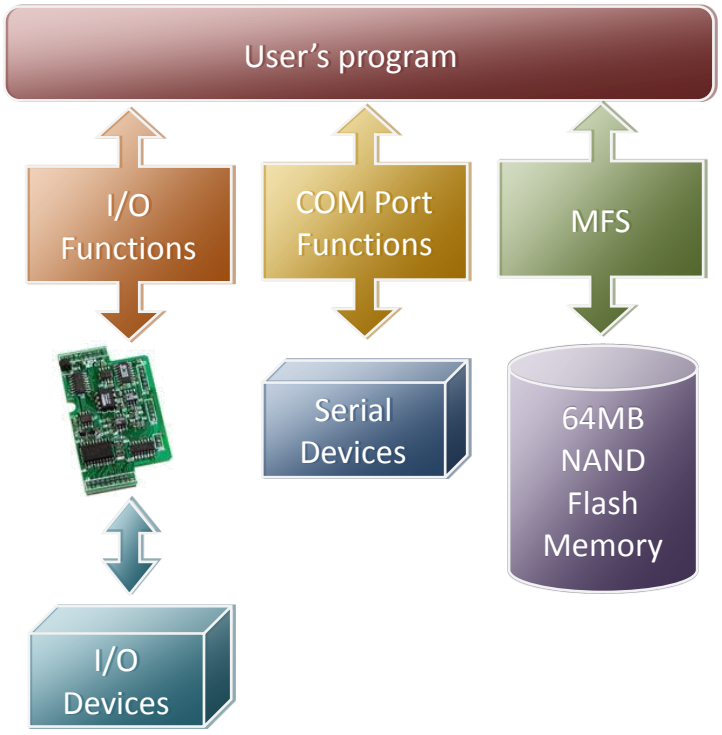
Hardware Supported

µPAC-5000-FD (With 64MB Flash Memory), NVRAM: all of the 31 bytes.

Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

Applications

Log data with timestamp, Log data and forward via the Ethernet



MFS Specifications

Item	Description
Disk size	1/2 size of the flash memory size
File number	456 files max. for each disk
File size	Disk Size max. for each file
File name	12 bytes max (case sensitive)
File operation modes	1. Read only 2. Write only: Creates a new file to write data, or overwrite a file (if the file is already exit). 3. Append: appends data to a file.

Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

File handle	10 max for each disk. For read mode: the 10 file handles can all be used for reading operation on each disk. Total 20 files can be opened for reading mode. For write and append mode: only 1 file handle can be used for writing operation on all disks.
Writing verification	Yes. Default is enabled. Calling mfs_EnableWriteVerification and mfs_DisableWriteVerification can change the setting.
Automate file system recovery	Yes. If an unexpected reset or power loss occurs, closed files, and files opened for reading are never at risk. Only data written since the last writing operation (mfs_WriteFile,) might be lost. When the file system reboots, it restores the file system to its state at the time of the last writing operation.
Writing speed	mfs_WriteFile: 147.5 KB/Sec (verification enabled) (default) 244.0 KB/Sec (verification disabled) mfs_Puts: 142.1 KB/Sec (verification enabled) (default) 229.5 KB/Sec (verification disabled)
Reading speed	mfs_ReadFile: 734.7 KB/Sec mfs_Gets: 414.2 KB/Sec
Max. length of writing data	32767 bytes.
Max. length of reading data	32767 bytes.

Resources upload

- MFS SDKs:
  http://ftp.icpdas.com/pub/cd/8000cd/napdos/upac-5000/demo/basic/lib/
- MFS Demos:
  http://ftp.icpdas.com/pub/cd/8000cd/napdos/upac-5000/demo/basic/64mb_flash/

Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

Appendix D. More C Compiler Settings

This section describes the setting of the following compilers:

- Turbo C 2.01 Compiler
- BC++ 3.1 IDE
- MSC 6.00 Compiler
- MSVC 1.50 Compiler

Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

D.1. Turbo C 2.01

You have a couple of choices here, you can:

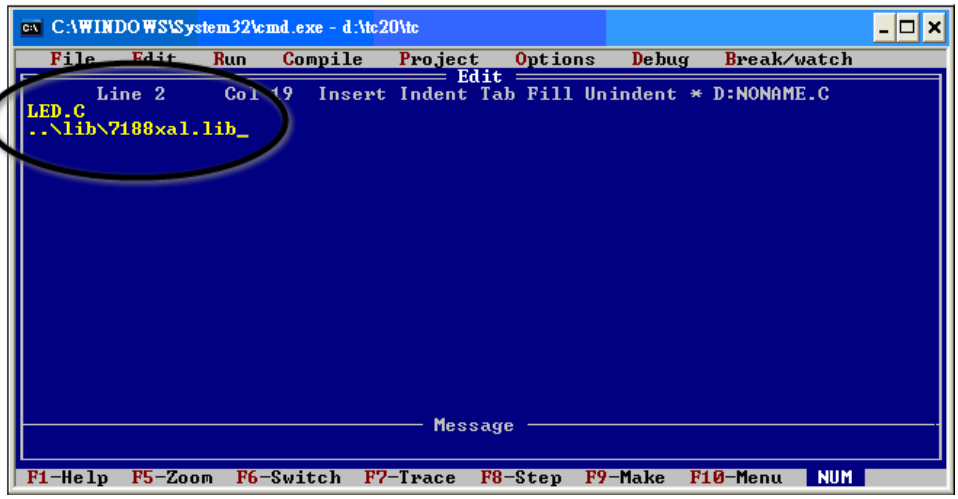
1: Using a command line

For more information, please refer to
CD:\8000\NAPDOS\8000\841x881x\Demo\hello\Hello_C\gotc.bat
tcc -Ic:\tc\include -Lc:\tc\lib hello1.c ..\..\Demo\basic\Lib\uPAC5000.lib

2: Using the TC Integrated Environment

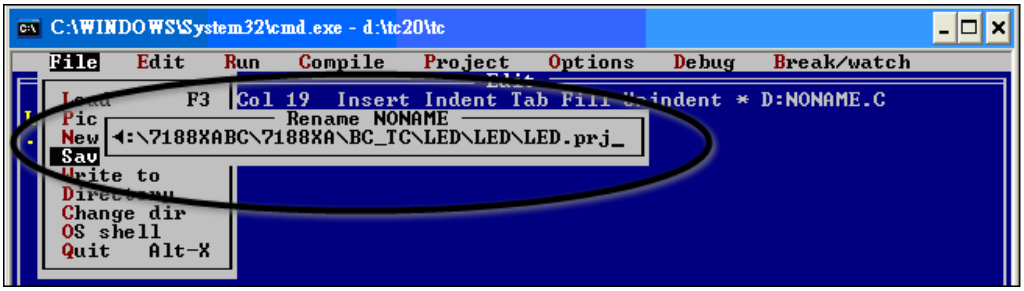
Step 1: Executing the TC 2.01 Step 2: Editing the Project file

Adding the necessary library and file to the project

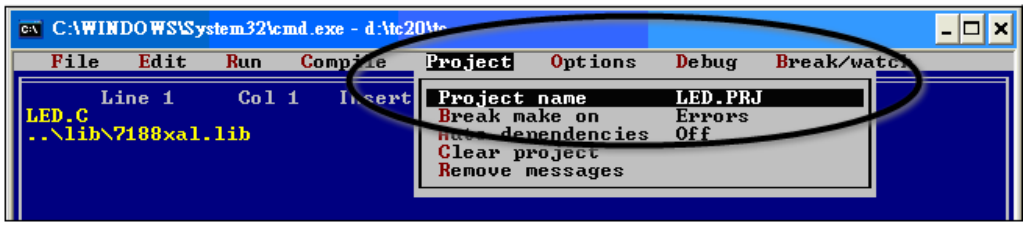

C:\WINDOWS\System32\cmd.exe - d:\tc20\tc
File Edit Run Compile Project Options Debug Break/watch
Edit
Line 2 Col 19 Insert Indent Tab Fill Unindent * D:NONAME.C
LED.C
..\lib\7188xal.lib_
Message
F1-Help F5-Zoom F6-Switch F7-Trace F8-Step F9-Make F10-Menu NUM

Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

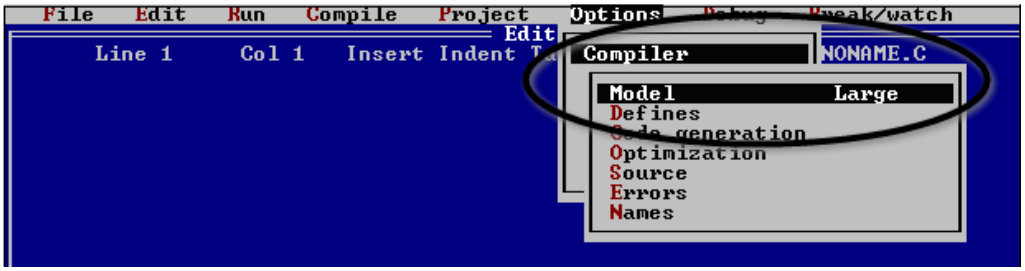
Step 3: Save the project and entering a name, such as LED.prj

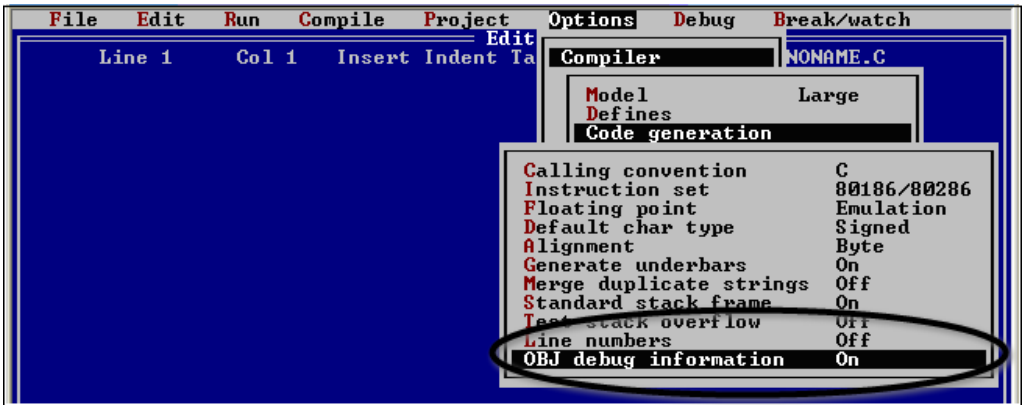


Step 4: Load the Project



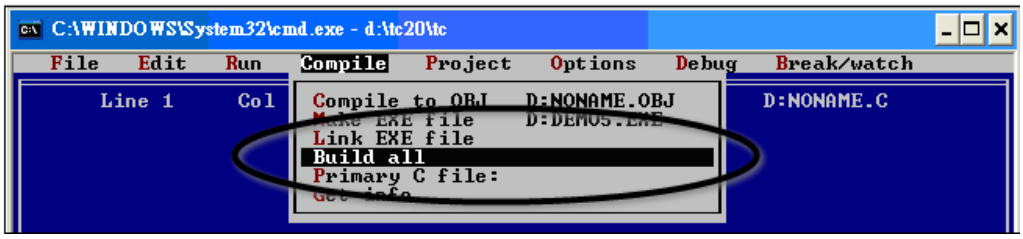
Step 5: Change the Memory model (Large for uPAC5000.lib) and set the Code Generation to 80186/80286

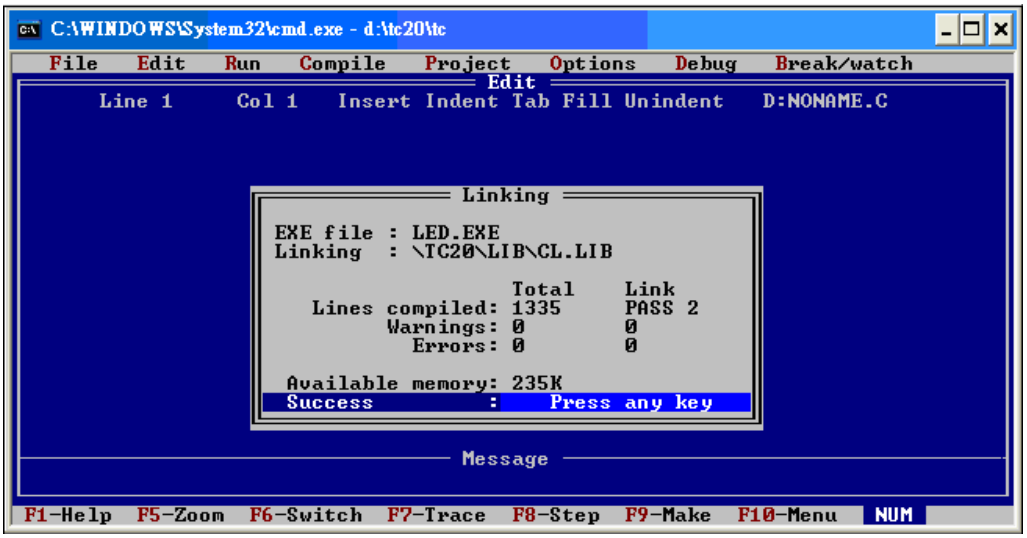




Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

Step 6: Building the project



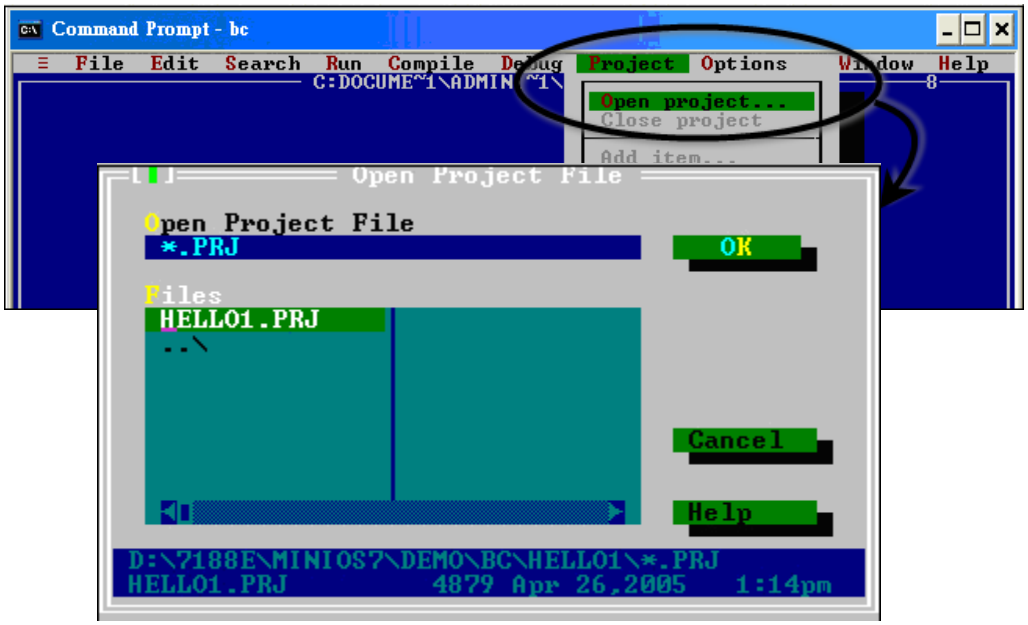

C:\WINDOWS\System32\cmd.exe - d:\tc20\tc
File Edit Run Compile Project Options Debug Break/watch
Edit
Line 1 Col 1 Insert Indent Tab Fill Unindent D:NONAME.C
Linking
EXE file : LED.EXE
Linking : \TC20\LIB\CL.LIB
Total Link
Lines compiled: 1335 PASS 2
Warnings: 0 0
Errors: 0 0
Available memory: 235K
Success : Press any key
Message
F1-Help F5-Zoom F6-Switch F7-Trace F8-Step F9-Make F10-Menu NUM

Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

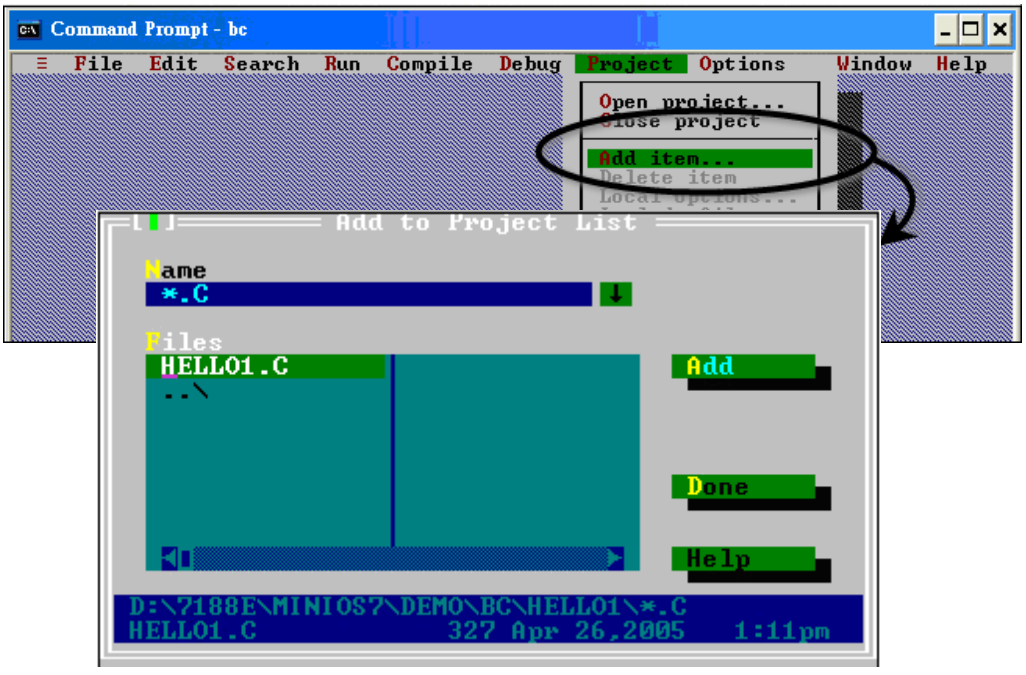
D.2. BC++ 3.1. IDE

Step 1: Executing the Borland C++ 3.1

Step 2: Creating a new project file (*.prj)

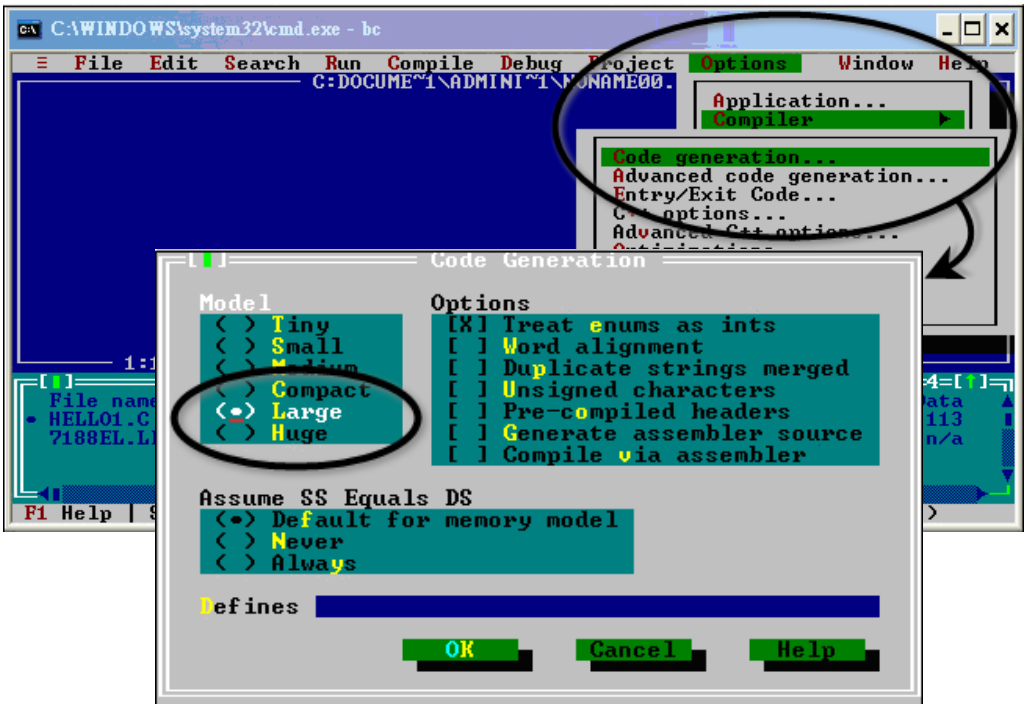


Step 3: Add all the necessary files to the project

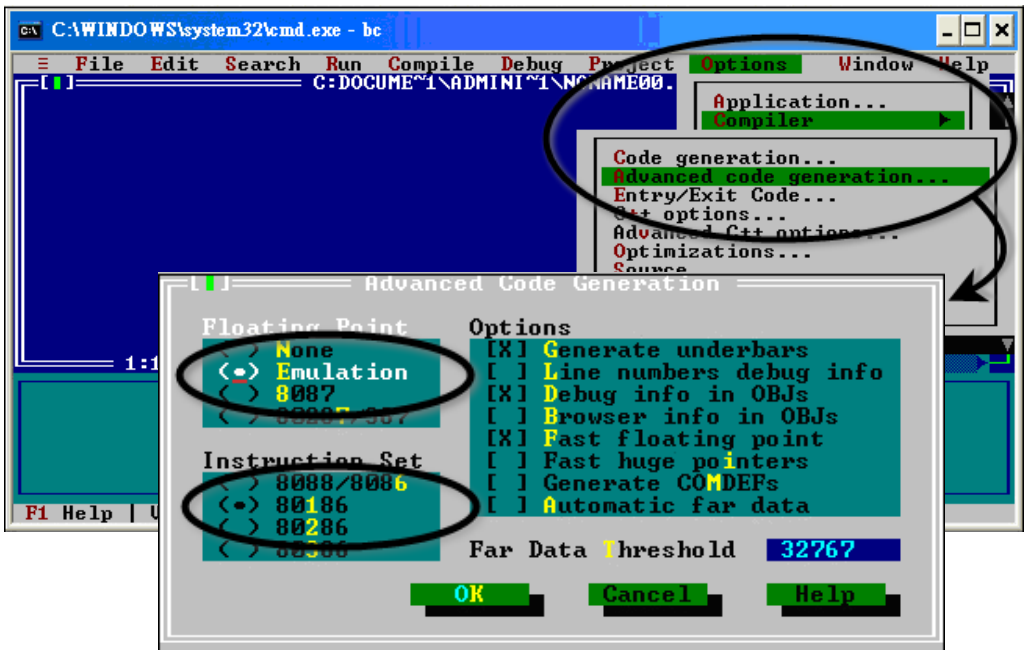


Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

Step 4: Change the Memory model (Large for uPAC5000.lib)

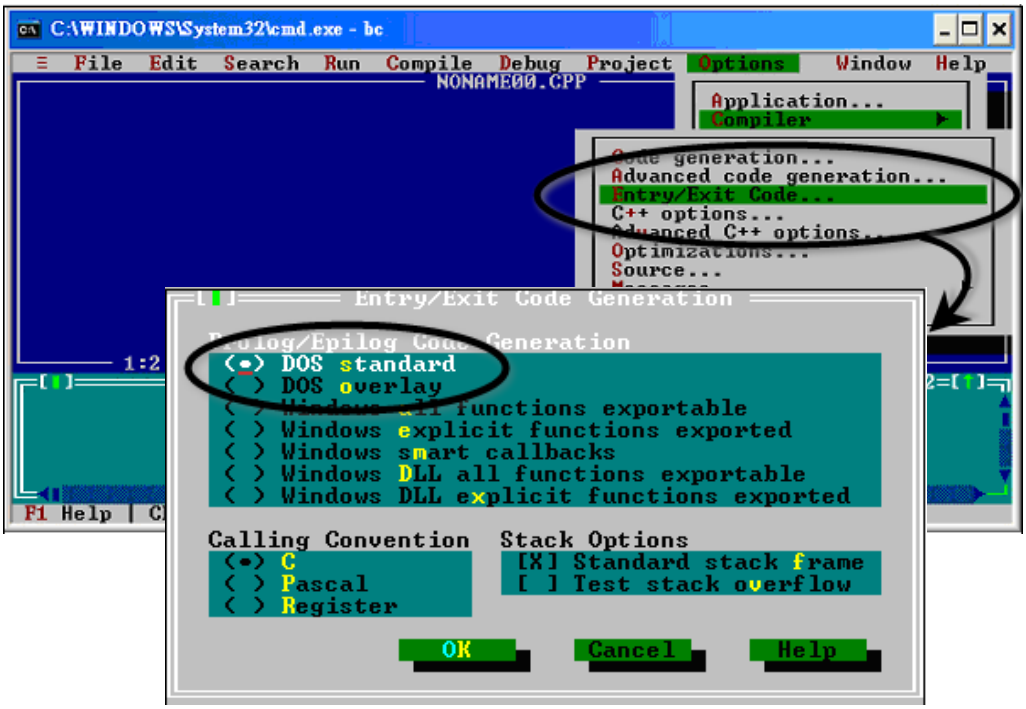


Step 5: Set the Advanced code generation options and Set the Floating Point to Emulation and the Instruction Set to 80186

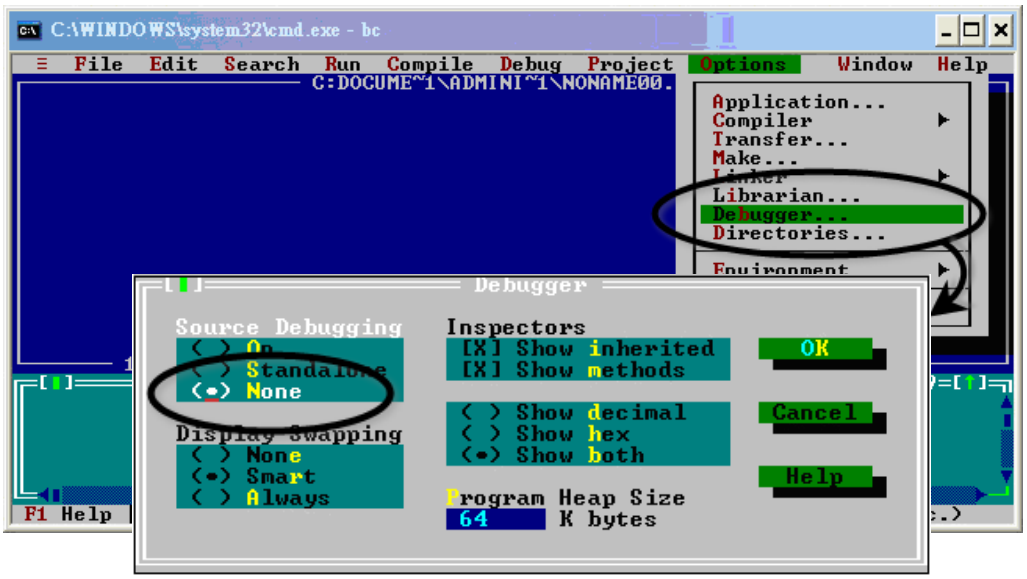


Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

Step 6: Set the Entry/Exit Code Generation option and setting the DOS standard

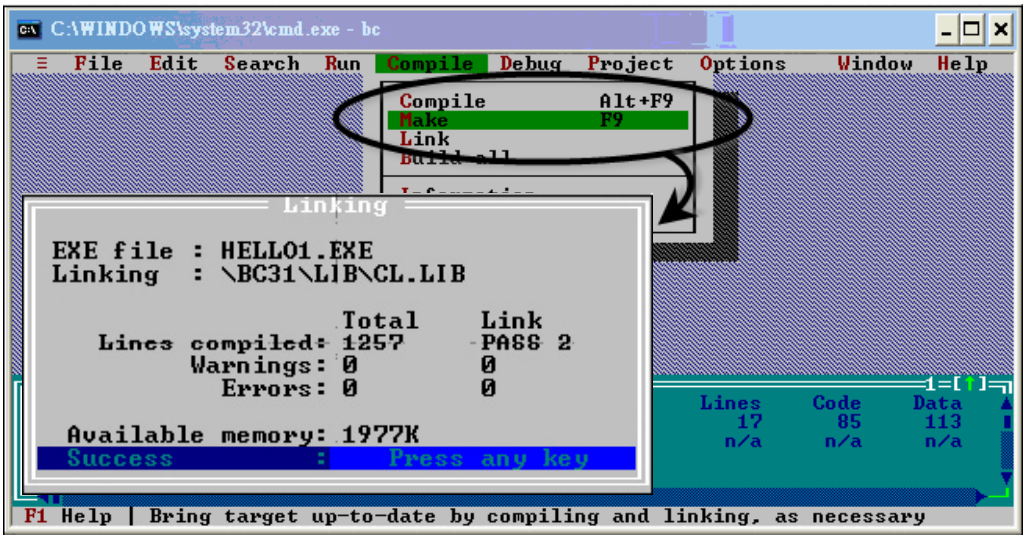


Step 7: Choosing the Debugger…and set the Source Debugging to None



Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

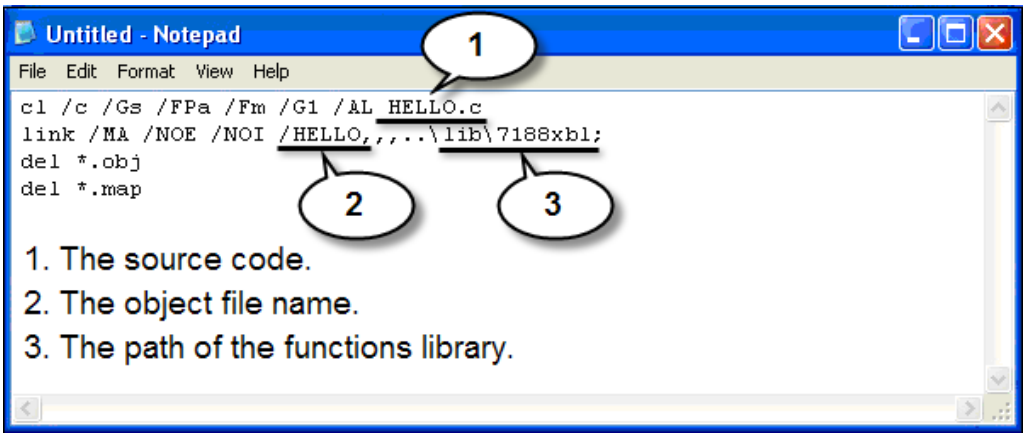
Step 8: Make the project



Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

D.3. MSC 6.00

Step 1: In the source file folder, create a batch file called Gomsc.bat using the text editor



Tip & Warnings

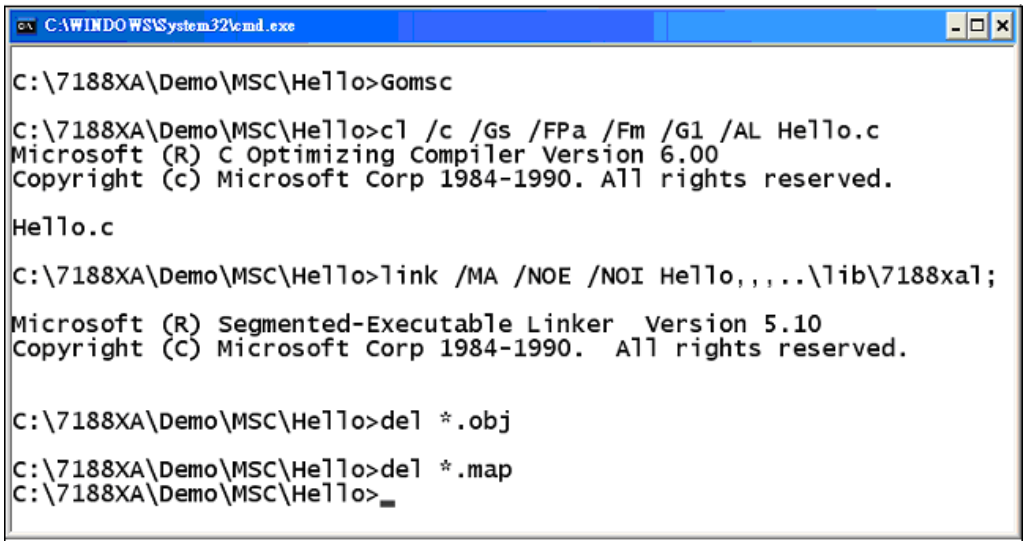
/C Don't strip comments /GS No stack checking

/Fpa: Calls with altmath /Fm [map file]

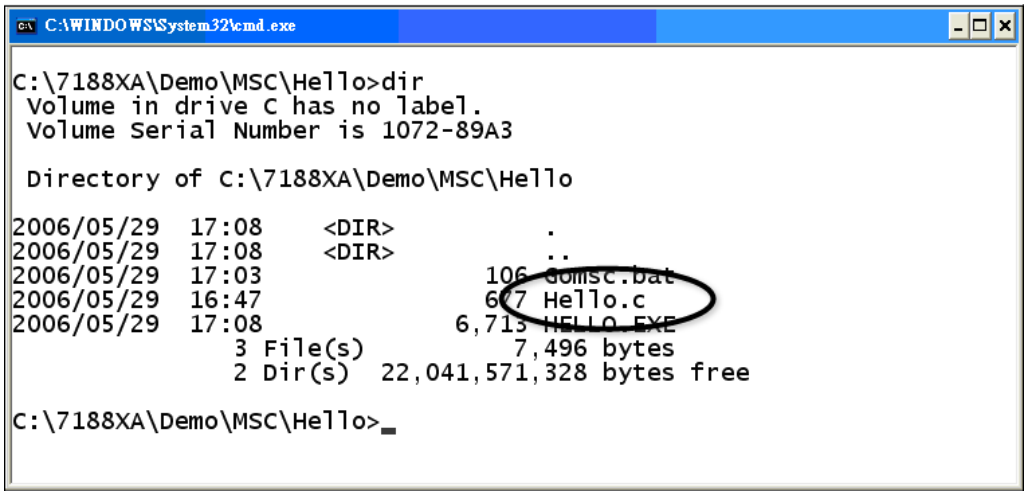
/G1: 186 instructions /AL Large model

Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

Step 2: Run the Gomsc.bat file


C:\WINDOWS\System32\cmd.exe
C:\7188XA\Demo\MSC\Hello>Gomsc
C:\7188XA\Demo\MSC\Hello>cl /c /Gs /FPa /Fm /G1 /AL Hello.c
Microsoft (R) C Optimizing Compiler Version 6.00
Copyright (c) Microsoft Corp 1984-1990. All rights reserved.
Hello.c
C:\7188XA\Demo\MSC\Hello>link /MA /NOE /NOI Hello,,,,..\lib\7188xal;
Microsoft (R) Segmented-Executable Linker Version 5.10
Copyright (C) Microsoft Corp 1984-1990. All rights reserved.
C:\7188XA\Demo\MSC\Hello>del *.obj
C:\7188XA\Demo\MSC\Hello>del *.map
C:\7188XA\Demo\MSC\Hello>

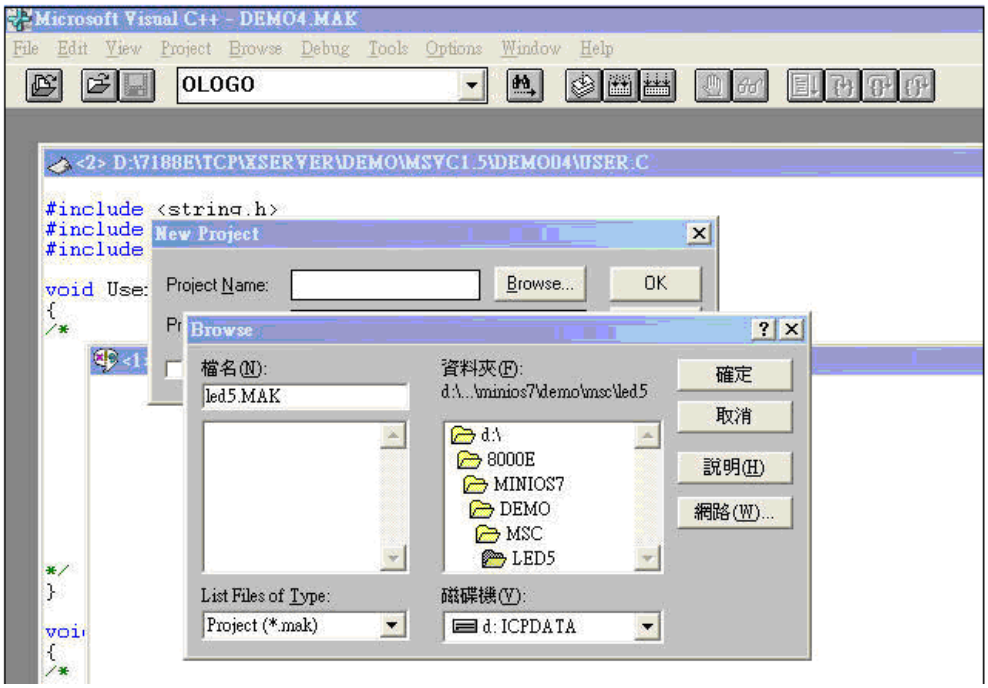
Step 3: A new executable file will be created if it is successfully compiled



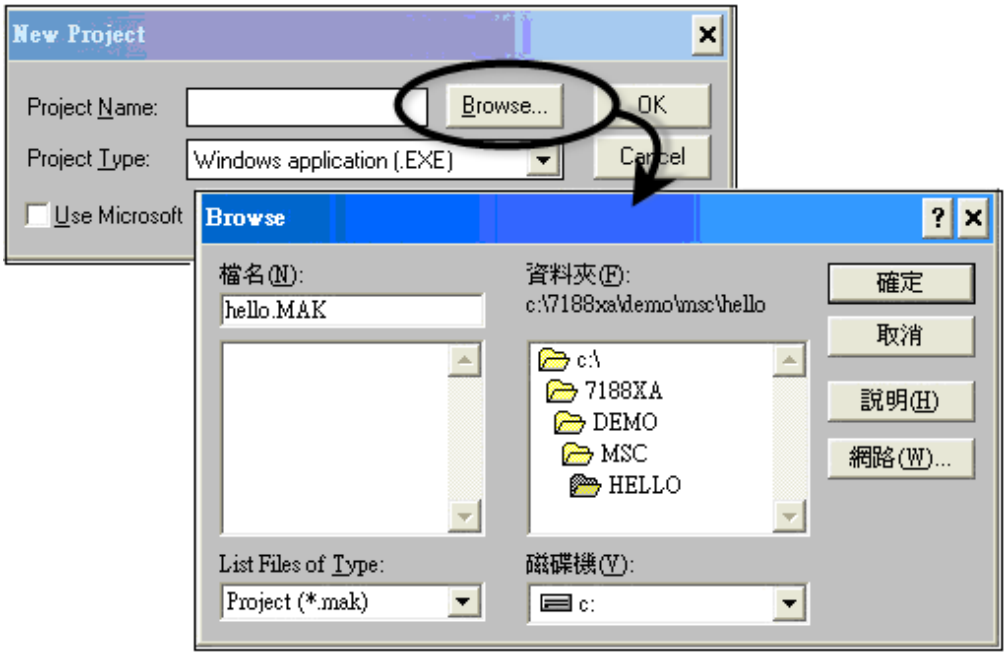
Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

D.4. MSVC 1.50

Step 1: Run MSVC.exe

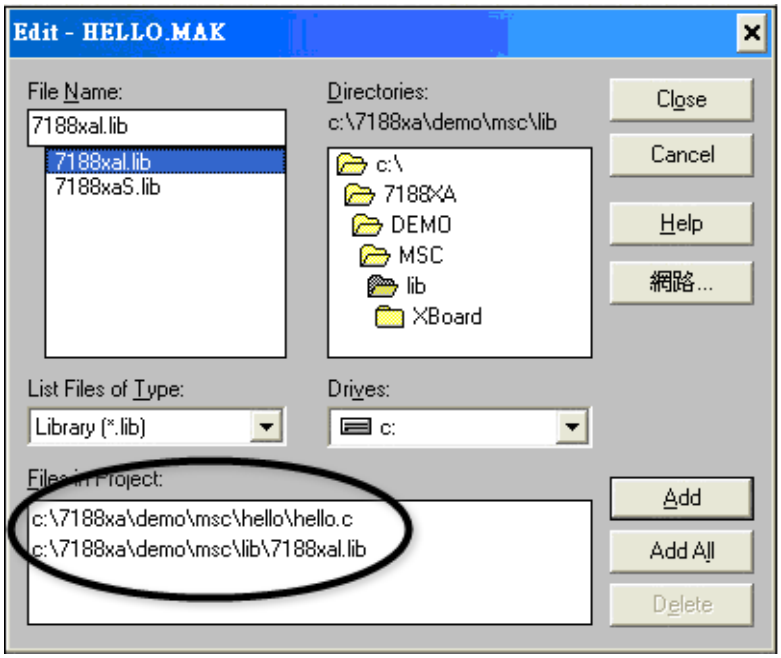


Step 2: Create a new project (*.mak) by entering the name of the project in the Project Name field and then select MS-DOS application (EXE) as the Project type

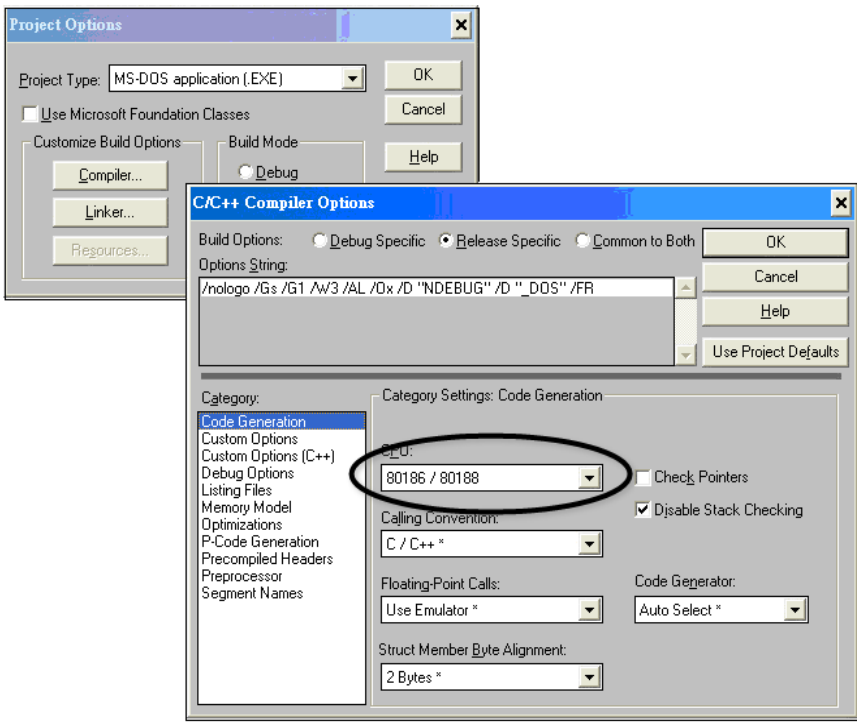


Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

Step 3: Add the user's program and the necessary library files to the project

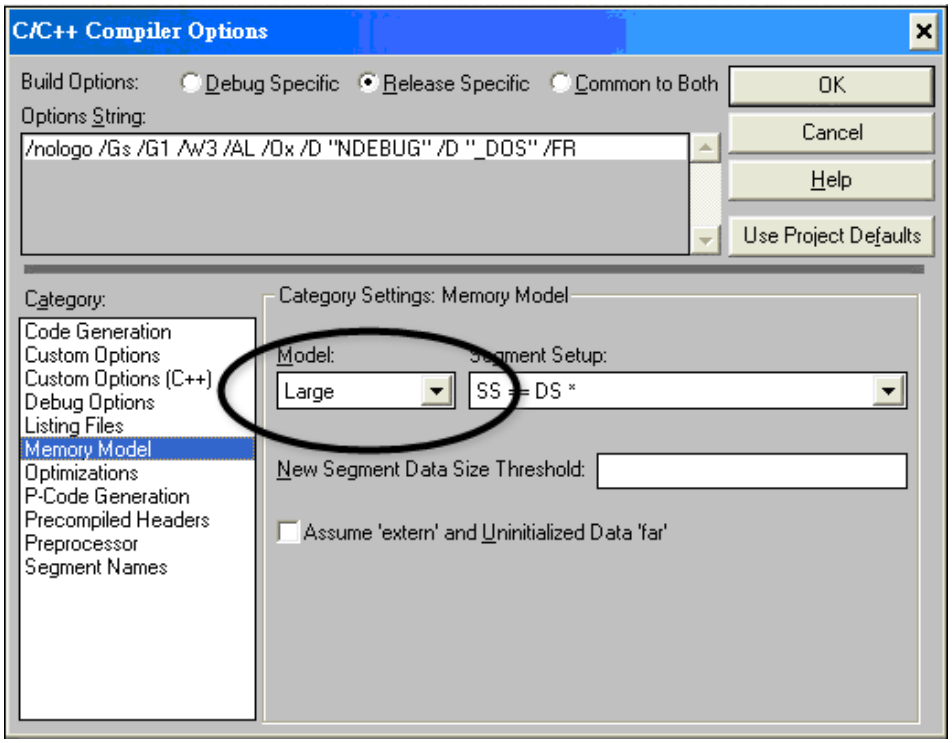


Step 4: Set the Code Generation on the Compiler.

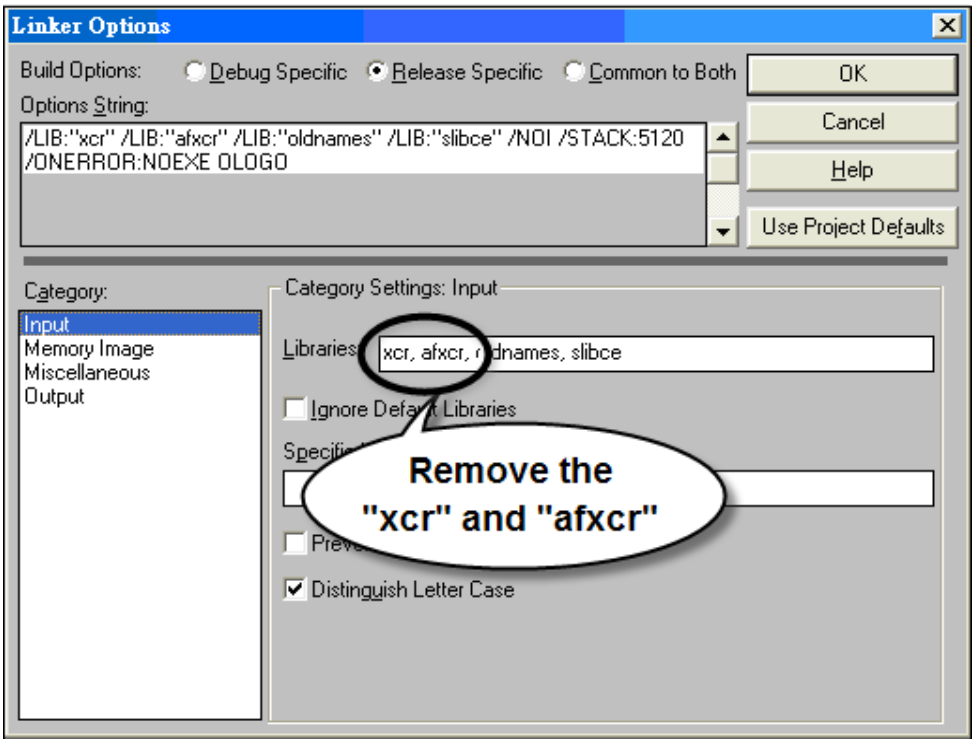


Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

Step 5: Change the Memory model (large for uPAC5000.lib)

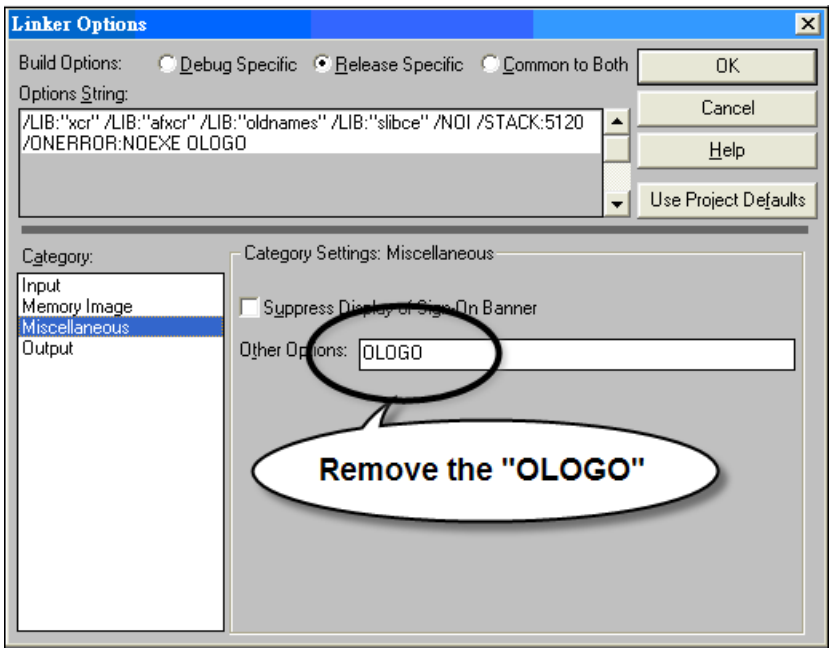


Step 6: Remove the xcr, afxcr library from the Input Category

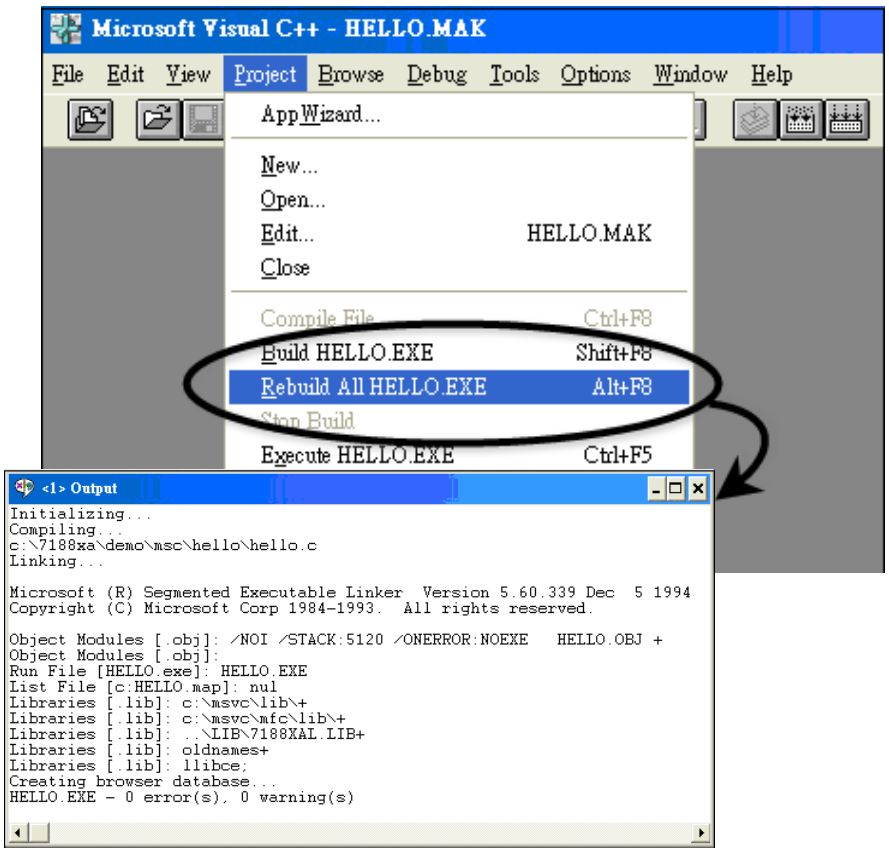


Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com

Step 7: Remove the OLOGO option from the miscellancous Category.



Step 8: Rebuild the project



Copyright © 2010 ICP DAS Co., Ltd. All Rights Reserved. E-mail: service@icpdas.com